Maria Nelson de Lemos Correia

Ferramentas para Análise de Posicionamento de Pessoas em Centros Comerciais

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Setembro de 2013

Maria Nelson de Lemos Correia

Ferramentas para Análise de Posicionamento de Pessoas em Centros Comerciais

Relatório de estágio submetido à Faculdade de Ciências da Universidade do Porto como parte dos requisitos para a obtenção do grau de Mestre em Ciência de Computadores

Orientador: Dr. Roberto Colazingari
Co-orientador: Prof. Doutor Alípio Jorge

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Setembro de 2013

Para todos aqueles que gostam e que não gostam de mim, aqueles que me apoiaram e aqueles que não estiveram por perto.
Obrigada por serem como são!

Agradecimentos

De entre as muitas pessoas que merecem os meus agradecimentos por terem acompanhado e apoiado o desenvolvimento deste estágio, gostaria de destacar, em primeiro lugar, o meu co-orientador Professor Doutor Alípio Jorge, pela devoção profissional e em especial pela partilha de saber e conhecimento. Agradeço ao meu orientador Dr. Roberto Colazingari, pela paciente ajuda e destemida orientação. Gostaria também de agradecer aos elementos da empresa Around Knowledge em especial os Drs. Fernando Freitas, Luis Correia, Ilídio Silva, Jorge Gonçalvez, José Vieira e Hugo Silva, pela ajuda disponibilizada sempre que esta foi solicitada.

Não posso terminar sem um agradecimento especial ao Dr. Celso Ferreira pela paciência, carinho e tempo disponibilizado.

Resumo

Os centros comerciais (CCs) começam a enfrentar um novo paradigma devido às alterações dos hábitos de consumo. Os consumidores portugueses tendem a voltar ao comércio tradicional, o que paralelamente diminui o fluxo de pessoas nos CCs.

Os comerciantes esforçam-se por encontrar formas de adquirir conhecimento sobre os seus usuais ou potenciais clientes de modo a com eles fomentar uma boa relação, com a finalidade de aumentar as receitas e promoção da lealdade dos consumidores. Atualmente estão disponíveis algumas ferramentas que permitem a recolha, organização e análise do fluxo de pessoas em determinados pontos, que vão dos mais simples e baratos como um torniquete, aos mais complexos e caros, por exemplo as câmaras de video que captam informação (*video-analytics*). Estas apresentam algumas falhas, encontrámos em alguns casos erros associados à contagem efetuada ou ao facto de não apresentarem os dados em tempo real.

Conscientes destes problemas a Around Knowledge (AK)[1] decidiu criar um produto que colmate as mencionadas lacunas. Surgiu então o *Business Intelligence Positioning System* (BIPS) como ferramenta na área de estudos de mercado. Para a recolha de informação, o BIPS, utiliza os sinais de radiofrequência emitidos por dispositivos móveis que se encontram no espaço em estudo.

A AK realizou um teste piloto onde os dados estudados foram recolhidos numa grande superfície comercial com mais de 70 mil metros quadrados. Neste estágio explorámos os dados de localização dos dispositivos e desenhámos e implementámos uma API (*Application programming interface*) que permite receber os dados dos dispositivos, tratá-los e enviá-los para uma Base de Dados (BD) que será acedida pela interface gráfica (IG). Para tentar responder às perguntas de negócio relevantes explorámos técnicas de análise de dados e algoritmos de *data mining*.

Este documento apresenta o estudo efetuado, os algoritmos implementados e os seus

[1] www.aroundknowledge.com

resultados para uma amostra de dados, bem como a integração na ferramenta de análise construída pela empresa.

Palavras chave: *Data Mining*, Análise de Comportamento, Análise de dados, Localização de dispositivos móveis.

Abstract

Shopping centers (SCs) are now facing a new paradigm due to changes in consumer habits. Portuguese consumers tend to return to traditional commerce, which simultaneously decreases the flow of people in SCs.

Traders strive to find ways to gain knowledge about their usual or potential customers in order to encourage them with a good relationship, aiming to increase revenues and promote consumer loyalty.

Currently there are some tools available which enable the collection, organization and analysis of the flow of people in certain points. From the simplest and cheaper, as a tourniquet, to the more complex and more expensive, for example by capturing information using video cameras (video analytics) . The tools which enable the collection, organization and analysis of the flow of people have some flaws. Among these flaws we can find some error associated with the counts made or the the difficulty of obtaining the data in real time.

Aware of these problems Around Knowledge (AK) [1] decided to create a product to fill in the gaps mentioned. Business Intelligence Positioning System (BIPS) sprung forward as a tool in the field of market research. To collect information, BIPS uses radio frequency signals emitted by mobile devices that are in the space under study.

AK conducted a pilot test wth data collected from a shopping centre with more than 70 thousand square meters. During this internship we explored the location data of the devices and we have designed and implemented algorithms for an interface. This interface enables the collection of position data and their treatment. The results are sent to a database that will be accessed by the GUI (Graphical user interface) . To provide answers to the relevant business questions we explored data analysis techniques and data mining algorithms.

This report presents the study we carried out, the implemented algorithms, the tests

made and the integration in the analysis tool built by the company .

Key words: Data Mining, Behavioral Analysis, Data Analysis, Mobile devices location.

Acrónimos

AK	Around Knowledge
BD	Base de Dados
BIPS	*Business Intelligence Positioning System*
CC	Centro Comercial
Crisp-DM	*CRoss-Industry Standard Process for Data Mining*
DBI	*DataBase Integration*
GSM	*Global System Mobile Communications*
GSP	*Generalized Sequential Patterns*
IG	*Interface Gráfica*
KDD	*Knowledge Discovery in Databases*
PrefixSpan	*Projected Sequential Patterns*
RJDNC	*R Java Database Connectivity*
rss	*Received Signal Strength*
RTLS	Sistema de Localização em Tempo Real
SEMMA	*Sample, Explore, Modify, Model, Assess*
SGBD	Sistema de Gestão de Base de Dados
SPAM	*Sequential Pattern Mining*
SPMF	*Sequential Pattern Mining Framework*
Weka	*Waikato Environment for knowledge Analysis*
XML	*eXtensible Markup Language*

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

O bem estar e a satisfação do cliente são, cada vez mais, as prioridades dos comerciantes, conhecer o mercado e as necessidades dos clientes é uma preocupação de pequenos e grandes lojistas. No que respeita ao cliente, a vantagem é do pequeno comerciante. Bom exemplo disso é o da mercearia, onde há uma relação de proximidade proporcionando uma previsão das necessidades da procura. Por outro lado os grandes comerciantes, tais como os gestores de um centro comercial (CC), não têm a possibilidade de saber/conhecer essa informação. Podemos por isso concluir que uma melhor relação com o cliente permite fidelização por parte deste, consequentemente existindo um aumento de lucros.

Nas últimas décadas foram desenvolvidas formas de recolher, organizar e analisar o fluxo de pessoas em determinados pontos, que vão dos mais simples e baratos como, um torniquete ou a contabilização manual de pessoas, a outros mais complexos e caros como, através da interrupção de dois feixes paralelos de raios de infravermelhos, ou ainda através de câmaras de video que capturam a informação (*video-analytics*). Estas plataformas têm algumas falhas - erro associado às contagens, problemas em apresentar os dados em tempo real ou levarem à aglomeração de pessoas em certos pontos - e por isso a Around Knowledge (AK) [1] (figura 1.1) decidiu criar um produto que permita contornar estes obstáculos.

A AK, uma empresa de consultadoria informática, com vista a ajudar à tomada de decisão dos gestores de grandes empresas de comércio a retalho, avançou com um projecto na área de estudos de mercado. Surgiu o *Business Intelligence Positioning System* (BIPS), uma plataforma que permite contagens mais precisas e acesso

[1] www.aroundknowledge.com

à informação em tempo real, mais barato que plataformas equivalentes. O BIPS utiliza radiofrequência para tentar determinar a posição de dispositivos móveis que atravessam um determinado perímetro. Com essa informação é possível a utilização de técnicas de Análise de Dados para a deteção de padrões úteis que permitam ajudar os gestores a tomarem decisões lucrativas.

Neste estágio estendemos o BIPS com um conjunto de ferramentas de exploração e análise de dados de localização.

Figura 1.1: Logo da empresa Around Knowledge (AK)

1.1 Objectivos

Milhares de pessoas circulam diariamente num CC e em lojas, os seus percursos podem ser registados e analisados de forma a fornecer aos seus gestores informação relevante para os seus negócios. O BIPS consegue recolher dados através de âncoras que detetam dispositivos por rádiofrequência e, seguindo o percurso destes dispositivos, obtêm-se dados que após o tratamento e análise apoiam a decisão do gestor. O BIPS estima com menor erro o número de visitas que são efetuadas num determinado espaço delimitado identificando o trajeto efectuado pelo visitante, tornando-o por isso mais fiável do que as restantes plataformas.

A análise exploratória dos dados espaço-temporais, através de técnicas de estatística e de *Data Mining*, permite identificar padrões e tendências que podem ajudar os gestores nas suas decisões.

Neste estágio fizemos o desenho e a implementação de algoritmos para uma API de exploração e análise de dados de localização. Estes algoritmos permitem o tratamento de dados enviados pelas âncoras (dispositivo que agrega um micro-controlador e uma antena para a deteção dos dispositivos móveis) e o envio dos resultados para uma BD. Em particular, responder às seguintes questões:

1. Qual o número de visitas por dia, hora, zona, piso ou acesso?
2. Qual o número de visitantes a passar na praça da alimentação?
3. Quais as cinco lojas mais visitadas?
4. Qual a loja escolhida como primeira paragem pela maioria dos clientes?
5. Qual o tempo gasto numa visita?
6. Qual o tempo médio passado a fazer compras (*ShoppingTime*)?
7. Qual o tempo médio de passeio (*DwellTime*)?
8. Qual o tempo despendido na primeira paragem?
9. Qual o tempo despendido por loja?
10. Qual o caminho mais utilizado pelos clientes a partir de um acesso?
11. Qual o caminho mais utilizado pelos clientes?

O objetivo deste estágio foi a implementação de um conjunto de ferramentas que permitam apoiar o gestor a encontrar respostas para as questões apresentadas. Estas foram desenvolvidas utilizando, como caso de estudo, os dados recolhidos num grande CC do norte de Portugal. Importa mencionar que, atualmente, o conjunto de ferramentas implementadas estão já integradas no BIPS que se encontra em fase de produção.

1.2 Estrutura deste documento

Este documento encontra-se dividido em sete partes, descrevendo desde o estudo efetuado à implementação do conjunto de ferramentas. O capítulo 1 fornece uma breve descrição sobre o estágio. O capítulo 2 apresenta a metodologia seguida, as ferramentas e tecnologias utilizadas no decorrer deste estágio e ainda algoritmos de *data mining* estudados. O capítulo 3 apresenta duas plataformas comerciais equivalentes ao BIPS, a sua estrutura e situação atual e a comparação entre os produtos. O capítulo 4, por sua vez refere-se à compreenção e tratamento dos dados. Já o capítulo 5 expõe a modelação por questão proposta pelo cliente. O capítulo 6 apresenta partes do produto final a ser comercializado pelo BIPS. Como último capítulo 7 temos uma breve conclusão sobre o trabalho efetuado.

Capítulo 2

Ferramentas e Tecnologias Utilizadas

Para o desenvolvimento deste estágio foi necessário definir as estratégias a seguir e escolher ferramentas a utilizar. Estudaram-se possíveis metodologias e ferramentas para proceder à expansão do BIPS.

Com base em critérios definidos pela AK, como por exemplo a escolha prévia do Sistema de Gestão de Base de Dados (SGBD) Cassandra, foram escolhidas as várias ferramentas utilizadas. As suas funções vão desde o armazenamento dos dados, interligação entre todas as partes do projeto, transferência dos dados à BD, ao estudo das possíveis soluções para a criação da ferramenta.

A metodologia de *data mining* escolhida e as ferramentas de desenvolvimento e implementação, bem como a sua utilização serão descritas neste capítulo.

2.1 Metodologia de *Data Mining*

Nas componentes do nosso trabalho em que fizemos análise de dados, seguimos a metodologia *CRoss-Industry Standard Process for Data Mining* (Crisp-DM), modelo de processo que define abordagens, normalmente utilizadas por especialistas de *Data Mining*, com o intuito de resolver problemas[Cor11]. Após o estudo de mais duas metodologias, *Knowledge Discovery in Databases* (KDD) e *Sample, Explore, Modify, Model, Assess* (SEMMA), foi escolhido este processo. Esta escolha recaiu sobre a metodologia Crisp-DM devido ao resultado do estudo comparativo efetuado por A. Azevedo e M. F. Santos [AS08] e por ter sido proposta por um consórcio que inclui várias grandes e pequenas empresas, universidades e especialistas de *Data Mining*.

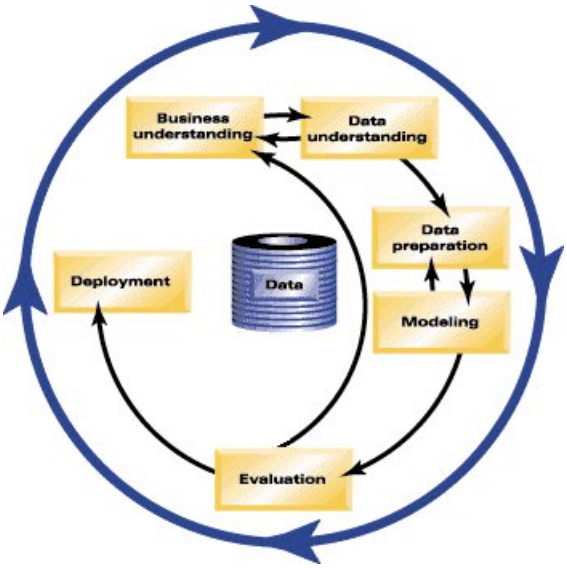

Figura 2.1: Ciclo Crisp-DM

Este processo consiste num ciclo de seis fases (Figura 2.1):

- Fase 1 - *Business Understanding* - Compreensão do negócio;
- Fase 2 - *Data Understanding* - Compreensão os Dados;
- Fase 3 - *Data Preparation* - Preparação dos Dados;
- Fase 4 - *Modelling* - Modelação;
- Fase 5 - *Evaluation* - Avaliação;
- Fase 6 - *Deployment* - Implementação e Execução.

A sequência das fases não é rígida, permitindo a movimentação livre entre fases.

1. ***Business Understanding*** É o momento em que se determina os objetivos do negócio, quais os requisitos necessários, os riscos e contingências e o custo/benefício do negócio, e avalia a situação. determinamos a meta fundamental do *Data Mining* e procedemos à avaliação inicial de ferramentas e técnicas a utilizar.

2. ***Data Understanding*** Inicía-se a recolha dos dados e a exploração e verificação da sua qualidade, detetam-se os conjuntos de dados curiosos dando, desta maneira, resposta a algumas questões interessantes do negócio.

3. ***Data Preparation*** Após a fase anterior, de recolha de dados, os mesmos serão trabalhados e estudados, permitindo verificar aqueles que se irá utilizar no processo de *Data Mining* e/ou aqueles que irão ser excetuados, existindo uma justificação admissível para essas exclusões, em suma, esta fase passa por todas as atividades que permitem construir o conjunto de dados finais a partir dos dados em bruto.

4. ***Modelling*** São selecionadas e aplicadas várias técnicas de modelação, os seus parâmetros são testados e o modelo avaliado.

5. ***Evaluation*** Procede-se à avaliação dos resultados e, tendo em conta o negócio em vista, analisa-se os critérios utilizados de forma a entender se os mesmos foram bem sucedidos. Após esta análise são aprovados os modelos cujos critérios triunfaram.

6. ***Deployment*** Organização dos dados obtidos nas fases anteriores, de forma a que o cliente os possa utilizar e compreender.

2.2 Base de Dados

Para guardar os dados vindos do CC, foi necessário escolher uma BD. A AK, em detrimento dos SGBDs relacionais, apostou nos SGBD não relacionais, estes são cada vez mais numerosos e mais utilizados. A sua principal característica é não possuírem relações entre as entidades. Após o estudo de alguns SGBDs não relacionais - MongoDB, Cassandra e o HBase - a AK optou pela utilização do Cassandra como método de armazenamento de dados.

O Cassandra é um SGBD não relacional [ver figura 2.2] que organiza os dados por *Keyspaces* - podem ser consideradas BDs num SGBD relacional - e *Column Families* - correspondendo a uma tabela num SGBD relacional. A escolha deveu-se, em grande parte, ao facto de ser um sistema cujas operações de escrita são mais rápidas que as de

leitura[1], uma vantagem quando se pretende a análise de grandes quantidades de dados. Finalmente, não devemos esquecer que este SGBD se encontra em constante evolução. Realçar ainda que o facto deste SGBD permitir particionamento, aumentando a capacidade de replicação dos dados, foi também uma razão para a sua escolha[Cas]. Para guardarmos os dados neste SGBD temos de colocá-los em *Keyspaces*, estas têm um conjunto de *Column Families* e, dentro das últimas temos *Columns* - registos - estas *Columns* são compostas por um nome e um valor [ver figura 2.2].

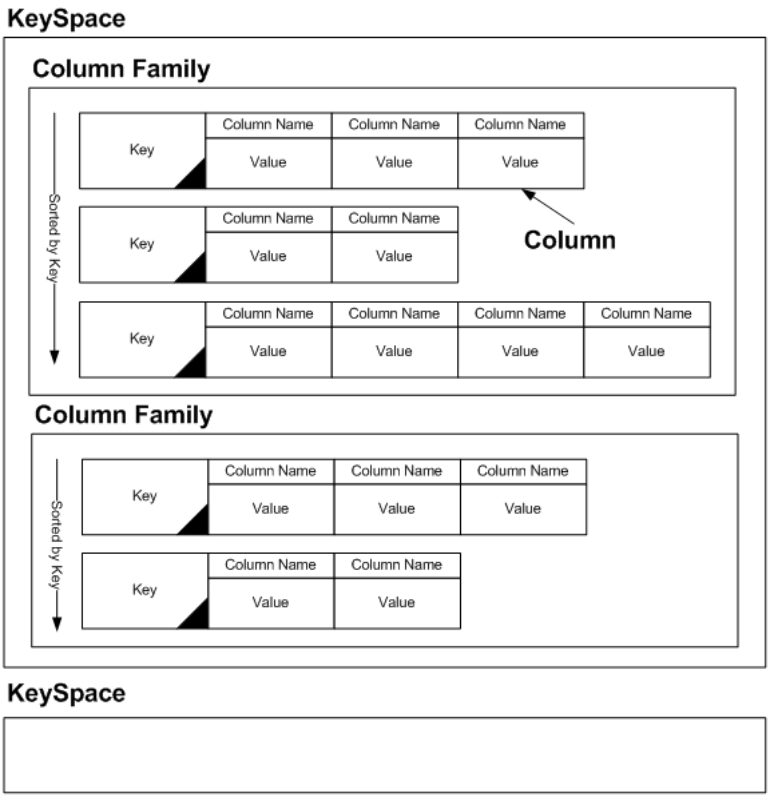

Figura 2.2: Estrutura do Cassandra

[1] Quando são limitadas pela velocidade do disco

2.3 Linguagens de Programação

2.3.1 Python

O Python é uma linguagem de programação de fácil aprendizagem e eficaz no que se refere a programação orientada a objetos. A sua sintaxe e natureza de interpretação torna-o uma linguagem ideal para *scripting* e desenvolvimento rápido de aplicações em muitas áreas, bem como na maioria das plataformas.

Esta linguagem foi escolhida como ferramenta para o tratamento dos dados, *i.e.*, para aceder à *keyspace* no SGBD Cassandra e transformar os dados obtidos de forma a facilitar a análise dos mesmos.

Foram várias as razões que levaram à sua escolha, destacamos: a sua fácil utilização; possuir bibliotecas que permitem a ligação ao Cassandra de uma forma rápida e limpa; ser de utilização livre e ser portável.

A escolha da biblioteca para aceder ao Cassandra, levantou um novo problema - foi necessário proceder à análise da biblioteca *lazyboy*[Git12a] - inativa há mais de dois anos - e a *pycassa*[Git12b], esta última foi a nossa escolha.

2.3.2 R

O R é um ambiente e linguagem de programação que é utilizado para computação estatística, disponibiliza um conjunto de comodidades integradas para a manipulação de dados, cálculos e exibição gráfica, permitindo uma extensa variedade de técnicas de análise estatística.[GIea97]

RCassandra e RJDBC

Tanto o RCassandra como o *R Java Database Connectivity* (RJDBC) são pacotes disponibilizados para a linguagem R, o seu objetivo é a ligação a SGBD's.

O RCassandra providencia uma ligação direta às funcionalidades mais básicas do Cassandra, tal como *login*, *updates* e pesquisas diretas[Urb12a, Urb12b], o RJDBC é um pacote que implementa uma *Database Integration* (DBI[2]).

[2]Componente de software que permite que uma aplicação Java interaja com uma BD.

Um dos requisitos necessários do RJDBC é a utilização da linguagem de programação Java, as duas formas de acesso foram testadas, caindo a escolha sobre o RCassandra. Esta deveu-se, em grande parte, ao facto de esta forma de ligação ser mais rápida.

2.4 *Data Mining*

2.4.1 WEKA - *Waikato Environment for Knowledge Analysis*

O WEKA foi desenvolvido na Universidade de Waikato na Nova Zelândia[3], trata-se de uma coleção de algoritmos de *machine learning*, dirigidos a tarefas de *Data Mining*. Os algoritmos podem ser aplicados diretamente a um conjunto de dados a partir do nosso próprio código[MHW09].

2.4.2 RapidMiner

O RapidMiner é um dos lideres mundiais em sistemas *open source* para *Data Mining*, encontra-se disponível como uma aplicação autónoma para análise de dados e contem um motor de *Data Mining* para a integração de produtos próprios. Utiliza esquemas de aprendizagem, avaliadores de atributos da ferramenta WEKA e esquemas de modelação estatística da linguagem R.

Disponibiliza uma IG que gera um arquivo *eXtensible Markup Language* (XML) que define os processos analíticos que o utilizador deseja aplicar aos dados, alternativamente, o motor pode ser chamado a partir de outros programas ou usado como uma aplicação e as funções individuais podem ser chamadas a partir da linha de comandos.[RI]

2.4.3 SPMF - *Sequential Pattern Mining Framework*

O SPMF é uma plataforma *open source*, escrita em Java por Philippe Fournier-Viger, que fornece a implementação de 52 algoritmos de *Data Mining*:

- deteção de padrões sequenciais (*sequential pattern mining*);

[3] www.waikato.ac.nz

- formulação de regras de associação (*association rule mining*);
- deteção de conjuntos de itens frequentes (*frequent itemset mining*);
- deteção de regras sequenciais (*sequential rule mining*);
- *clustering*.

Pode ser utilizado como um programa independente através de uma interface simples ou através da linha de comandos, permitindo a integração do código de cada um dos algoritmos noutros programas ou aplicações.[FV13a]

2.4.4 Algoritmos de Deteção de Padrões - *Sequential Mining Patterns*

Um dos muitos problemas do *Data Mining* é a descoberta de sequências frequentes a partir de uma BD. O objetivo é descobrir sequências de eventos frequentes, conceito introduzido por Agrawal e Srikant. No seu modelo uma BD é um conjunto de transações, cada transação é um conjunto de itens e está associado a um identificador de cliente e um identificador de tempo[AS95]. Se procedermos à ordenação dos dados por identificador de cliente e depois por identificador de tempo, obtemos um conjunto de sequências de clientes, cada sequencia de cada cliente mostra as suas ações por ordem de acontecimento.

De uma forma geral, o problema de deteção de sequências frequentes, implica descobrir subsequências que são, de certo modo, frequentes entre todas as sequências dos clientes. Foram, por isso, estudados alguns dos algoritmos criados com o intuito de resolver o problema de qual o caminho mais frequente.

GSP - *Generalized Sequential Patterns*

Entre os algoritmos mencionados, o GSP[SA96] é considerado um dos mais eficientes, este algoritmo procede à consulta da BD um determinado número de vezes.

Durante uma iteração (i), são encontradas sequências de tamanho i mais frequentes, desta forma o número de consultas efetuadas à BD é determinado pelo comprimento das sequências mais frequentes e mais longas da mesma.

Se a BD for grande e se contiver sequências muito compridas entre as mais frequentes, o custo computacional deste algoritmo é elevado.

Algorithm 2.4.1: GSP($DB, Support, ItemSet$)

procedure GSP($DB, Support, ItemSet$)
- $C_1 \leftarrow$ {({item}) | item ϵ ItemSet}
- Scan DB to get support(sup) of every sequence in C_1
- $L_1 \leftarrow$ {s | s ϵ C_1, sup(s) >= support}
- $i \leftarrow 1$
- **while** $L_i! = 0$
  - **do**
    - $C_{i+1} \leftarrow GSP\_Gen(L_i)$
    - Scan DB to get support(sup) of every sequence in C_{i+1}
    - $L_{i+1} \leftarrow$ {s | s ϵ C_{i+1}, sup(s) >= support}
    - $i \leftarrow i + 1$
- **return** $(L_1 \cup L_2 \cup ... \cup L_{i-1})$

Seguindo o pseudo-código do GSP (Algoritmo 2.4.1), sabemos que este algoritmo recebe três parâmetros, uma BD, um valor de suporte mínimo ($Support$) e um conjunto de itens ($ItemSet$).

Em primeiro lugar, pega em cada item i do conjunto de itens $ItemSet$ e forma uma sequencia candidata, ($\{i\}$), de tamanho 1 (C_1), faz a primeira consulta à BD e verifica todas as sequências candidatas de tamanho 1 para obter L_1 que contém todas as sequências de comprimento 1 que têm suporte não inferior a Support. L_1. É gerado o próximo conjunto de sequências C_2 através da aplicação da função GSPGen a L_1. Esta função gera sequências candidatas de tamanho $i + 1$ considerando todas as sequências frequentes de tamanho i, é efetuada uma nova consulta à BD para que todas as sequências frequentes de tamanho 2 sejam encontradas, obtendo assim L_2. Esta geração e verificação de candidatos é efetuada até que não se encontrem mais sequências frequentes, finalmente será retornada a sequencia mais frequente $L_1 \cup L_2 \cup ... \cup L_{i-1}$.

Verificamos que se o tamanho da sequencia frequente mais longa na BD for de n, o algoritmo terá de proceder à consulta da BD, pelo menos, n vezes[MZC01]. Este algoritmo tem, independentemente da implementação ser ou não detalhada, três custos

inerentes:

- Pode gerar um conjunto de sequências candidatas bastante elevado, mesmo para um conjunto de dados moderado.
- O comprimento de cada sequencia candidata aumenta por um a cada consulta à BD. Generalizando, para encontrar um padrão sequencial de comprimento l, o algoritmo terá de fazer a consulta à BD pelo menos l vezes, tal acarreta um custo bastante elevado quando existem padrões extensos.
- Quando procuramos entre longos padrões sequenciais, é gerado um número elevado de candidatos.

  Um padrão sequencial longo contem um número elevado de combinações de subsequências, e essas subsequências devem ser geradas e testadas, logo o numero de sequências candidatas é exponencial ao comprimento dos padrões sequências a serem pesquisados.

PrefixSpan - *Projected Sequential Pattern*

Para colmatar as desvantagens do algoritmo acima descrito, Pei et al. (2001) criaram o algoritmo PrefixSpan que consegue reduzir a criação de padrões candidatos a gerar para obter os frequentes. Este algoritmo segue os seguintes passos:

- **Input:** Base de dados S e o suporte mínimo, *Support*;
- **Output:** O conjunto completo de padrões sequenciais;
- **Método:** Chamar o prefixSpan($\langle\rangle$,0,S);
- **Sub-rotina:** prefixSpan(α,l,$S|_\alpha$);
- **Parâmetros:**

  α - padrão sequencial;

  l - tamanho de α;

  $S|_\alpha$ - se $\alpha \neq \langle\rangle$ então a BD projectada de α se não a BD S
- **Método:**

1. Consultar $S|_\alpha$ uma vez e encontrar o conjunto de b itens frequentes tal que:
   b pode ser reunido ao último elemento de α
   ou, $\langle b \rangle$ pode ser anexado a α
   e possa formar um padrão sequencial

2. Para cada item b frequente é necessário anexa-lo a α para formar um padrão sequencial α' e *output* α'
3. Para cada α', contruir a base de dados projetada $S|'_\alpha$ e chamar a função PrefixSpan com os parâmetros α', $l+1$ e $S|'_\alpha$

[PP01]

Algorithm 2.4.2: PREFIXSPAN($itemListS, Support, DB$)

$x \leftarrow$ Scan DB to get every item
$itemList \leftarrow$ {item | item ϵ {x where sup(item) >= Support}}
PREFIXSPAN($itemList, Support, DB$)

procedure PREFIXSPAN($itemList, Support, DB$)
for each $item \in itemList$
do
Form database $y|_{item}$
Find Supported Items
Prune database $y|_{item}$
if ($y|_{item}$ has more than 1 sequence)
then $z \leftarrow$ PREFIXSPAN($Items, Support, y|_{item}$)
$itemList \leftarrow$ **join**($item, Items + z$)
return ($itemList$)

SPAM - *Sequential PAttern Mining*

Este algoritmo, ao contrário dos anteriores, é mais eficiente quanto maiores os padrões sequenciais da BD. É utilizada uma estratégia de procura em profundidade para gerar sequências candidatas e vários tipos de mecanismos, são implementados para ajudar à redução do espaço de pesquisa.

Os dados são guardados utilizando uma representação vertical de *bitmap*, a qual permite uma compressão significativa[AFGY02].

Capítulo 3

Soluções para Análise de Comportamento e Contagem de Pessoas

A necessidade de conhecer os padrões de comportamento do consumidor é um dos pontos fulcrais para o sucesso de qualquer negócio, são poucas as soluções que permitem perceber certos aspetos no comportamento do consumidor, como o caminho tomado ou as lojas visitadas. As plataformas disponíveis no mercado, para esse fim, apresentam limitações na identificação e contagem dos consumidores.

Neste capítulo vamos descrever algumas das soluções tradicionalmente utilizadas para a análise de comportamentos e contagens de pessoas, dando especial atenção à plataforma desenvolvida pela AK - o BIPS. Estas plataformas disponibilizam informação aos comerciantes e gestores de grandes superfícies, permitindo-nos aumentar receitas e melhorar a rentabilidade, analisando o que acontece nos locais de venda, através da contagem de visitantes, monitorização de filas, custo de vendas, número de funcionários, marketing e outros dados de desempenho.

3.1 Produtos Comerciais Estudados

Existem atualmente sistemas para monitoração do movimento de clientes em centros comerciais e outros espaços, estes oferecem apenas informação sobre o número de pessoas que entram e saiem do local e o número de pessoas que aí estacionam. No

que respeita à contagem pelo número de carros estacionados a limitação prende-se ao facto de existir quem o utilize e não entre na superfície comercial, criando uma falsa leitura dos dados. Podemos também (tentar) perceber o movimento dos consumidores através de estudos de mercado tradicionais, procurando definir o perfil do consumidor através de inqúeritos. Este metodo, apesar de obter dados fidedignos, demora vários dias para obtenção dos dados e obriga à presença de recursos humanos no local para contactar com os inquiridos. Podemos também colocar mecanismos nas portas que contam as entradas. No entanto, o metodo é pouco fiável já que a mesma pessoa pode entrar e sair do CC no mesmo dia e não passar da área referente ao acesso/porta. Finalmente, como alternativa, temos os sistemas de análise de vídeo que, apesar de serem mais fiáveis, tornam-se bastante mais dispendiosos e sofrem, também, de erros de contagem. Conseguimos por isso concluir que a percentagem de erro destas medições é grande, não trazendo nenhuma vantagem ao gestor.

3.1.1 Experian FootFall

O Experian FootFall é uma plataforma desenvolvida pela empresa Experian, esta empresa é lider mundial no fornecimento de informações, ferramentas analíticas e serviços de marketing a organizações e consumidores, através desta plataforma providencia serviços para os comerciantes e gestores de CCs com o propósito de aumentar os rendimentos analisando o que se passa nos locais de comércio. O Experian FootFall permite identificar o rácio de empregados/visitantes, nos momentos de pico, quanto tempo o visitante perde nas filas e qual o impacto disso nas vendas, quão eficazes são as equipas de venda a encorajar o visitante a voltar, qual a sua loja/área com a pior/melhor execução no que se refere a um conjunto de medidas de desempenho, onde o marketing é eficaz e em que tipo de lojas e qual o impacto de novos projetos e estratégias em diferentes locais. O FootFall utiliza, ainda, *software* para prever a pisada em cada local por forma a ajudar os gestores a planearem[Foo].

3.1.2 VMS - *Video Managment Software*

A 3VR é uma companhia de *"video intelligence"* (análise de vídeos) que analisa o registo das filmagens aumentando a segurança e prevenindo fraudes para melhor servir os consumidores. Criaram e comercializam o VMS. Este é um produto comercial que permite a contabilização de pessoas, através de deteção por câmaras, além da contabilização de pessoas, esta plataforma atinge outros propósitos como a vídeo

vigilância, vigilância facial, reconhecimento de matriculas, análise comportamental ou ainda análise de filas de pessoas.

Para os gestores de CCs, esta não é apenas uma plataforma de contabilização pois, permite também perceber quem é o seu cliente, fornece-lhe a noção da idade, sexo, comportamento do seu cliente, entre outros aspectos. Tudo isto através da combinação de análise e vídeos, integração em contexto e inteligência social de forma a entregar a informação vital que os comerciantes necessitam.

Permitindo-lhes distribuírem os seus produtos nas montras e/ou prateleiras, promoções a fazer e estratégias de gestão de filas. O comerciante tem acesso a uma aplicação que permite acesso às operações da loja em tempo real[3VR].

3.2 BIPS

O BIPS (Figura 3.1) é um Sistema de Localização em Tempo Real (RTLS), que permite procurar, seguir, gerir, analisar e trabalhar dados referentes à localização de indivíduos dentro de um edifício ou espaço aberto, desenvolvido pela AK. Esta plataforma permite, para além da elaboração de estudos de mercado, a contabilização de clientes. Ela consiste em estudar as rotas realizadas por um determinado conjunto de indivíduos e as suas tendências de orientação em tempo real, através de um sistema de posicionamento, em espaços delimitados, por radiofrequência.

O sistema baseia-se na instalação de dispositivos que, aqui denominados por âncoras, contêm uma antena para cada tecnologia, *Global System Mobile Communications* (GSM), Wi-Fi ou *Bluetooth*, posicionadas em locais específicos. Os dados obtidos são enviados para um servidor central, que guarda toda a informação posteriormente analisada detalhadamente e é realizado o registo das capturas (importa referir que todo este processo é realizado em tempo real).

Como podemos ver as potenciais aplicações vão de sistemas de segurança ao controlo de tráfego passando por usos comerciais[Kno13].

Figura 3.1: BIPS - *Business Intelligence Positioning System*

3.2.1 Situação Atual

O BIPS ajuda os comerciantes a perceberem o comportamento dos clientes, através da recolha de dados e disponibilização de informação que permite aos gestores a tomada de decisões informadas. Em geral estas plataformas têm a desvantagem de não fornecerem informação em tempo real e as que o conseguem fazer, não respondem a todas as circunstâncias.

Na maioria das soluções, a contagem das passagens dos visitantes é feita num único local. Com o BIPS os comerciantes conseguem informações sobre o cliente, de forma precisa e em tempo real, permitindo assim uma reação imediata.

O sistema da AK traz uma diminuição no erro das contagens e permite obter um leque maior de informação sobre os clientes que realmente chegam a entrar no estabelecimento. O BIPS fornece monitorização dos visitantes de forma anónima e coloca à disposição do comerciante relatórios em tempo real e de uma forma rápida, conveniente, precisa e menos dispendiosa que soluções equivalentes.

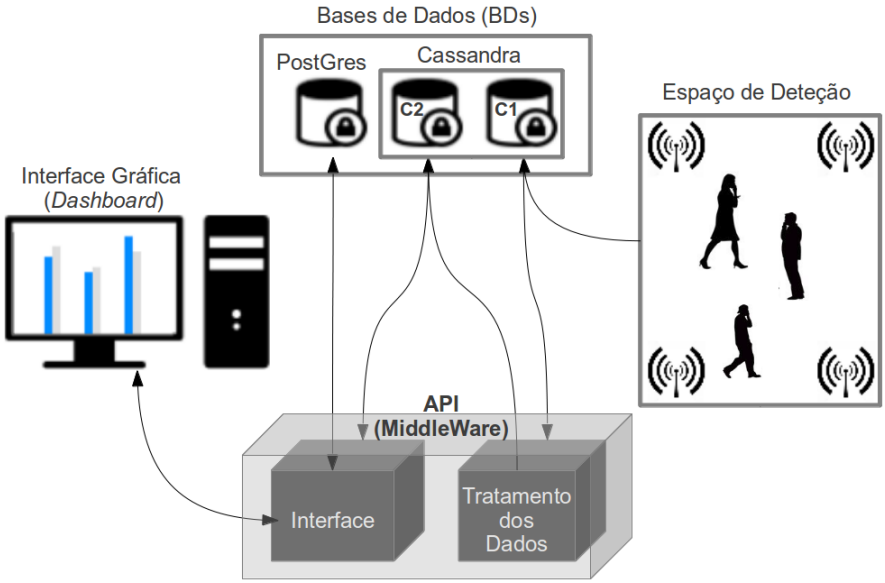

Figura 3.2: Estrutura do BIPS

No caso específico de um CC, o BIPS determina qual o fluxo de clientes, permitindo assim a otimização de rotas. Ajuda os gestores do CC na análise do resultado de eventos publicitários, monitoriza as zonas menos visitadas, prevê a movimentação de visitantes pelo CC e escôa zonas de maior congestionamento, permitindo aos visitantes a melhor experiência possível. Os comerciantes ao saberem as zonas de maior concentração de visitantes podem direccionar a atenção destes para certos produtos, áreas ou ofertas.

O objetivo final do BIPS é disponibilizar ao comerciante uma aplicação que lhe permita aceder à informação obtida a partir dos dados adquiridos pelas âncoras. O BIPS está estruturado da seguinte forma (Figura 3.2):

1. Dispositivos de deteção de radiofrequência;
2. Bases de Dados (BD);
3. Ferramenta de tratamento dos dados;
4. API[1] (Middleware);
5. Interface gráfica (IG).

3.2.2 Captura e Estrutura de Dados

As âncoras enviam os dados para uma *keyspace* em Cassandra (que se encontra na segunda parte da estrutura), que serão enviados para outra *keyspace* após a passagem pelo conjunto de ferramentas de tratamento de dados, esta última *keyspace* será acedida pela API, que enviará a informação à IG de forma a que o utilizador lhe possa aceder. As âncoras procedem ao envio de uma identificação do dispositivo[2] detetado, para um servidor, se nos 10 segundos seguintes o mesmo dispositivo for encontrado pelo mesmo conjunto de âncoras, a segunda deteção é ignorada. Este espaço de tempo garante que todos os dispositivos que têm de ser detetados serão mesmo detetados. Inicialmente guardaram-se os dados no servidor, anexando-lhes um *timestamp*, após o tratamento aos dados enviados pelas âncoras, foram anexadas as

[1] http://www.webopedia.com/TERM/A/API.html

[2] Seguindo as leis da Comissão Nacional de Proteção de Dados - CNPD - Lei nº67/98 de 26 de Outubro.

coordenadas, o grupo de âncoras e o piso em que o dispositivo foi detetado. Estes grupos são identificados por um conjunto de, pelo menos, três âncoras e os pisos são identificados através da força de sinal entre a âncora do piso um e o grupo de âncoras associadas no segundo piso. Os dados são enviados para uma *Keyspace* do SGBD Cassandra[Cas], e numa *Column Family*, são então inseridos por data (*Row Key*) e, à medida que estes vão chegando, são acrescentados todos os *timestamps* e dispositivos detetados nos 10 segundos seguintes, com a respetiva informação. Este estágio cingiu-se à segunda parte, ou seja, ao estudo e implementação de algoritmos que permitam criar um conjunto de ferramentas de tratamento de dados, dados estes que estão guardados numa *keyspace*.

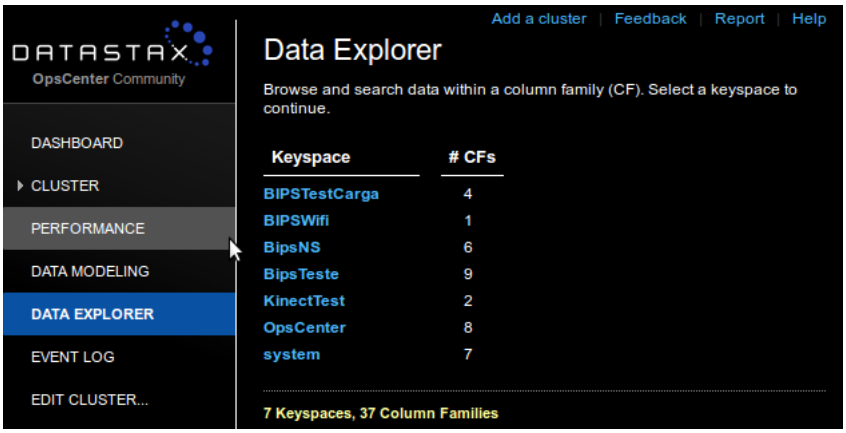

Figura 3.3: Vista do DataStax: todas as keyspaces guardadas no SGBD.

Foi criada uma *Column Family* que guarda toda a informação tendo como *Row Key* a data. Para cada data foi agregado um conjunto de *timestamps*, e para cada um destes foram anexados os dispositivos detetados nesses dez segundos e para cada dispositivo, o conjunto de âncoras em que este foi detetado.

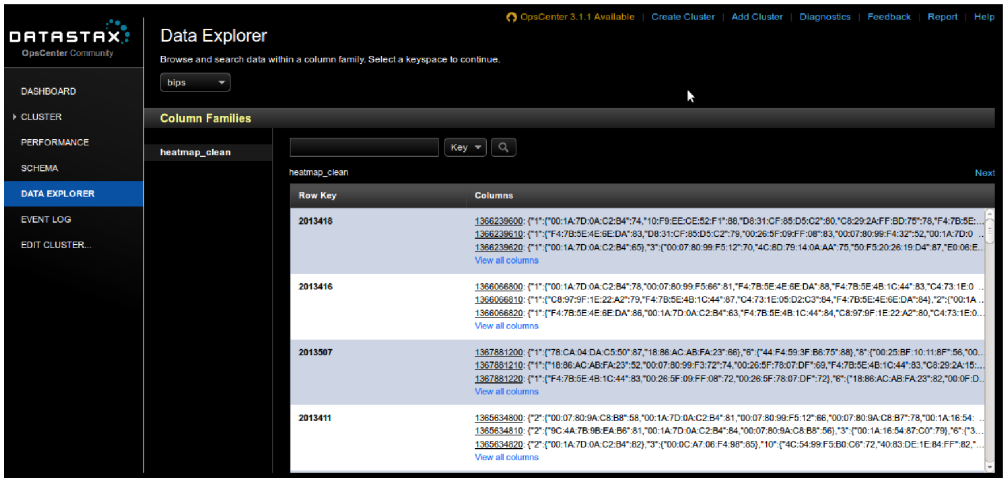

Figura 3.4: Base de Dados Final

3.2.3 Tratamento dos Dados

O tratamento de dados para as ferramentas de análise desenvolvidas neste estágio foi feito em Python, desde a sua aquisição, a partir do Cassandra, à sua utilização na resposta às perguntas efetuadas pelos gestores. Já sabendo como os dados são guardados no SGBD Cassandra, resta criar a ligação e proceder à sua aquisição. Os dados são guardados de forma sequencial, o que facilita o acesso quando apenas se pretende a análise de um determinado tipo de dados. A sua organização está relacionada com o objetivo principal deste estágio, a utilização dos algoritmos desenvolvidos para o tratamento dos dados. Mais à frente neste documento será abordada a implementação destes algoritmos. É de mencionar que foi feita uma tentativa de utilizar a ferramenta R e por isso, escolhida a biblioteca para proceder à ligação ao Cassandra mas, depois de alguns testes efetuados, devido a atraso quanto à resposta do Cassandra e envio dos dados para o R, esta ferramenta foi colocada de parte e suplantada pelo python.

3.3 Análise Comparativa dos Produtos

Sabemos que o FootFall é um dos concorrentes diretos do BIPS. Em primeiro lugar porque a maioria dos possíveis clientes da AK utilizam no momento esta plataforma e porque os dados disponibilizados pelo FootFall são equivalentes aos do BIPS. Sabemos

também que uma das mais valias dos sistemas de análise de vídeo é permitirem ter algum conhecimento sobre o cliente, como o sexo e a idade, algo que o BIPS não permite. Não podemos deixar de concluir que após a análise dos produtos o BIPS trouxe uma grande inovação ao mercado. Para além do seu potencial e precisão na informação que gera, preserva a privacidade dos utilizadores de equipamentos móveis, apesar de conseguir saber quais os passos que os mesmos dão.

Ao analisarmos os prós e os contras destas plataformas percebemos que são poucas as vantagens do FootFall e do VMS quando nos referimos a custo benefício. O BIPS é uma plataforma que permite acesso a informação equivalente, em tempo real, preservando a privacidade do cliente e a um custo mais baixo[3].

[3]Permite economizar até 89% do custo atual em relação aos métodos tradicionais de estudos de mobilidade. Esta percentagem foi conseguida após análise feita pela AK a valores de estudos de mercado bem como através de retorno por parte dosgestores do CC em estudo.

Capítulo 4

Compreensão e Preparação dos Dados

O objetivo deste estágio foi construir e integrar no BIPs um conjunto de ferramentas que permitem responder a algumas questões colocadas pelos gestores do CC, para isso foi necessário organizar ideias e definir conceitos, bem como estudar os dados para encontrar possíveis implementações. Depois dos conceitos e questões definidas analisamos os problemas para conseguir obter respostas, estudando quais os melhores algoritmos para resolver as questões e formas de apresentar os resultado. Neste capítulo vamos apresentar os estudos efetuados.

4.1 Visitas

Existem alguns fatores que definem uma visita: o dispositivo que a efetuou e a data e hora de inicio e de fim da mesma. Depois do tratamento dos dados, conseguimos obter um conjunto de visitas num dia, tendo a visita sido considerada o período de tempo em que o dispositivo se encontra a ser detetado por âncoras, o final da visita é definido pela ausência de deteções por, pelo menos, 3 horas (10.800 segundos). Por esta razão, e porque os clientes podem fazer mais do que uma visita no mesmo dia ao CC, um dispositivo pode ter mais que uma visita. Foi determinado o período de 3 horas porque o objetivo é definir visitas, totalmente distintas e não o esquecimento de algo, a ida ao parque de estacionamento ou ir fumar um cigarro.

Esta definição trouxe alguns problemas, os dispositivos que passem esse tempo nos cinemas seriam contabilizados como tendo efetuado duas visitas, não sendo isto, de facto, verdadeiro, foi necessário proceder à sua verificação. Por isso, se um dispositivo

desaparecer (deixar de ser detetado) por mais de cinco minutos perto do cinema e reaparecer no mesmo local passadas 3 horas, está implícito que esteve no cinema, o mesmo demonstrou ser verdade para quem faz compras na grande superfície anexada ao CC. É importante verificar se um dispositivo realizou mais do que uma visita ao CC.

Entre as perguntas a responder, encontra-se a questão "Qual o número de visitas por dia, hora, zona, piso ou acesso?". O número de visitas efetuadas num determinado dia é conseguido através da contagem das visitas encontradas, o tempo de visita é conseguido através da subtração do *timestamp* inicial ao final.

$$timestamp\ Inicial + Timestamp\ final = Tempo\ de\ Visita$$

Olhando para o exemplo 1 conseguimos saber que o tempo de visita deste cliente foi de 28 *minutos*[1].

Example 1 Um conjunto de deteções e *timestamps* associados[1]

Temos uma visita de um cliente ao CC e as respetivas associações entre âncoras e acessos.

$$\begin{aligned} pisos &\Rightarrow < piso1 : z1, piso1 : z2, piso2 : z3, piso2 : z4 > \\ zonas &\Rightarrow < z1 :< 1,2,3 >, z2 :< 4,5,6,7 >, z3 :< 8,9,10 >, \\ &\quad z4 :< 11,12,13,14 >> \\ detections &\Rightarrow < 1,2,3,5,6,7,7,7,7,7,7,7,9 > \\ timestamps &\Rightarrow < 1,3,17,18,19,20,21,22,23,24,25,26,29 > \end{aligned}$$

4.2 Acessos

Para se saber qual a porta de acesso e a de saída de cada dispositivo verificou-se qual a primeira e última âncora de deteção, posteriormente confirmou-se qual a associação existente entre as âncoras de deteção e os acessos obtendo o acesso por onde esse cliente entrou no CC nessa visita. Agregamos as visitas por acesso e procedemos à sua contagem.

[1] Tomando de principio que os *timestamps* estão em minutos.

Devido a alguns problemas demonstrados pelas âncoras (falhas na deteção), por vezes a primeira âncora não está associada a um acesso, quando tal acontece, verifica-se qual a última âncora e se esta estiver associada a um acesso, passa a ser esse acesso o de entrada.

Caso contrário é considerado um acesso desconhecido, i.e., se a âncora inicial não estiver associada a um acesso, será averiguado qual o acesso associado à âncora final e, no caso de esta não estar associada, a visita apenas será contabilizada para o número de visitas por acesso, já que a distribuição deste tipo de visitas será feita segundo a probabilidade de ter entrado, ou não, por uma determinada porta. O exemplo a ser estudado facilmente nos diz que o cliente entrou pela âncora 1 ou 7.

4.3 Zonas e Pisos

Inicialmente dividimos o CC em algumas secções que definimos como zonas, uma das questões que os gestores queriam ver respondida era "Qual o número de visitantes na praça da Alimentação?". Para responder a esta questão era necessário definir o espaço ocupado pela praça de alimentação, permitindo assim uma definição de zonas por todo o CC. As zonas, tal como os pisos, são defenidas no python através das âncoras, os pisos pouca explicação necessitam, se o CC tiver mais que um piso a identificação destes será o seu número, já que o CC, onde decorreu a recolha dos dados, tem mais que um piso, foi necessário associar as zonas a um piso. Após a definição das zonas e pisos, procedemos à contagem das visitas que passaram pelas âncoras e aquelas que passaram pelas zonas associadas a pisos. Conseguimos verificar, através do exemplo 1 que, durante esta visita o cliente esteve nas zonas $z1$,$z2$ e $z3$ e que esteve nos pisos $piso1$ e $piso2$. Através das condicionantes mencionadas acima, conseguimos respostas às questões 1,2 e 11 (Capítulo 1).

4.4 Entrada nas Lojas

Foi necessário definir as lojas e tentar encontrar as entradas dos clientes em determinada loja, esta tarefa foi um pouco complicada já que, para a maior feabilidade dos dados, seria necessário introduzir âncoras nas lojas, tal era impossível por concessão dos gestores do CC, houve então a necessidade de tentar outras vias para descobrir se um cliente teria entrado, ou não, numa determinada loja, o problema foi abordado de

duas formas diferentes:

- Incluir coordenadas da deteção do dispositivo. Bastava verificar se a posição do cliente estaria entre as coordenadas limítrofes da loja.
- Se o dispositivo deixar de ser detetado por cinco minutos (300 segundos) ou mais, significa que o cliente entrou na loja mais próxima à âncora em que desapareceu.

No que se refere à primeira solução a existência de coordenadas, como já foi mencionado em capítulos anteriores, impôs que, pelo menos três âncoras, detetem o mesmo dispositivo no mesmo *timestamp*, tal implica que todas as deteções efetuadas por menos do que três âncoras sejam descartadas, ou seja, temos perda de informação. Decidimos pela segunda opção, conseguindo obter as estadias (permanência na mesma âncora) ou entradas numa loja (deixar de ser detetado por qualquer âncora durante, pelo menos, cinco minutos). Depois de definida a diferença entre entrar ou não numa loja, estamos prontos para responder às questões 3,4,7 e 8 (Capítulo 1).

Para isso foi necessário encontrar todas as lojas visitadas por todos os clientes e proceder à sua contagem, bastou percorrer as deteções de cada dispositivo. Se a distância temporal entre uma detecção e a seguinte for maior que cinco minutos (300 segundos), então guardamos a âncora referente à primeira deteção[2] e essa será a loja em que o cliente entrou, no caso de o cliente ter sido detetado pela mesma âncora durante um período igual ao mencionado é considerado que esteve dentro de uma loja. A loja será a que está associada à âncora que detectou o cliente por esse espaço de tempo, se observarmos o exemplo 1 conseguimos saber que as lojas visitadas por este cliente são as associadas às âncoras 2 e 7.

4.5 *Shopping Time* e *Dwell Time*

O *Shopping Time* refere-se ao tempo que um cliente se encontra dentro de lojas e o *Dwell Time* é o oposto, é o tempo que um cliente passa no corredor a passear ou a ir de uma loja para outra. Estes conceitos serão úteis para se conseguir responder às questões 5 - "Qual o tempo gasto numa visita?"e 6 - "Qual o tempo médio passado a fazer compras (ShoppingTime)?". No caso do *Shopping Time* foi necessário encontrar o tempo passado em todas as lojas e proceder à sua soma. No que se refere ao *Dwell Time*, bastou remover ao tempo total da visita o *Shopping Time*, se olharmos

[2] As âncoras estão associadas a lojas.

novamente para o exemplo 1, podemos verificar que: O *Shopping Time* = 14 + 6 ou seja 20 *minutos*, enquanto que o *Dwell Time* = 28 − 20 o que perfaz 8 *minutos*. Conseguimos obter também os tempos por zona, na zona $z1$ foi efectuado um *Shopping Time* de 14 *minutos* e um *Dwell Time* de 2 e no piso 1 temos um *Shopping Time* de 20 *minutos* e um *Dwell Time* de 5.

4.6 Pré-processamento de Caminhos

Os gestores do CC solicitaram, como características da aplicação, a identificação e apresentação dos caminhos mais utilizados pelos clientes. Primeiro foi necessário perceber o que eram, para os gestores, os caminhos mais utilizados. Seria aquele que mais passagens tem?, ou seriam os caminhos por onde um maior número de visitantes passa?, i.e., refere-se ao momento em que um visitante passa de uma zona para outra e, sempre que faz esse percurso, ser contabilizado?, ou ser apenas contabilizado uma vez, independentemente do número de vezes que aí passa? Após reunião com os gestores entendemos que o objetivo era encontrar o caminho que é utilizado por um maior número de visitantes.

A definição informal de um caminho efetuado por uma pessoa é o conjunto de locais por onde essa pessoa passou. No caso em estudo, um caminho é constituído pela sequência temporal de deteções feitas por um conjunto de âncoras.

Após uma recolha inicial de dados, concluímos que existiam âncoras isoladas, no meio de um caminho definido. Ou seja, mesmo localizando-se mais distantes do dispositivo, as âncoras conseguem encontrá-lo com mais precisão do que as mais próximas, dado que existem interferências externas, como por exemplo, antenas 3G ou até mesmo quiosques multimédia distribuidos pelo local. Desta forma os caminhos tiveram de passar por uma limpeza antes de ser encontrado o mais utilizado. Seguidamente descrevemos este algoritmo que permite o pré-processamento dos caminhos.

4.6.1 Algoritmo Utilizado

Do estudo dos algoritmos anteriormente mencionados e de alguns testes efetuados concluímos que o PrefixSpan seria o mais indicado para o problema em causa, "*A definição do caminho mais frequentado*"[FV13b]. Deve-se ao tamanho dos caminhos não ser grande, o que, como mencionado é uma desvantagem para o SPAM, já que

a falta de tempo é significativa para proceder à implementação dos três algoritmos e algumas das ferramentas mencionadas no capítulo 2 permitem a sua utilização, os algoritmos foram testados com os dados obtidos da BD, através da utilização dessas ferramentas, o algoritmo GSP está implementado para o WEKA e RapidMiner, o RapidMiner demonstrou ter um erro no retorno desse algoritmo, problema que não se conseguiu resolver em tempo útil. O WEKA foi utilizado após a falha do RapidMiner ter sido detectada.

Decidimos correr a ferramenta SPMF, que permite verificar diferenças entre os algoritmos PrefixSpan e SPAM. Através da execução dos dois algoritmos com o mesmo conjunto de dados chegamos à conclusão que a utilizar um destes algoritmos o PrefixSpan seria o melhor, corroborando a conclusão obtida depois do estudo dos mesmo. O PrefixSpan em pouco tempo retornou soluções coisa que o SPAM não fez chegando a dar erro de memória. Para dar resposta ao problema, foi criado um processo de limpeza aos dados. Este processo é utilizado para cada um dos acessos para encontrar o caminho mais utilizado a partir de cada entrada.

Algorithm 4.6.1: PREPROCESSPATHS($path, AccessName$)

procedure PREPROCESSPATHS($path, AccessName$)
- **if** $path[0] == AccessName$
  - **then**
    - $p \leftarrow$ SMOOTHING($path$)
    - $p \leftarrow$ CHANGE(p)
    - $p \leftarrow$ REMOVEREPETITIONS(p)
    - $p \leftarrow$ TRANSFORM(p)
    - $preprocessedPaths \leftarrow preprocessedPaths + p$
- **return** ($preprocessedPaths$)

Como se aproximava a *"dead line"* para a entrega da aplicação aos gestores do CC, foi necessário proceder à implementação de um algoritmo que obtivesse resultados, como o algoritmo necessitava de algumas especificidades, os algoritmos estudados não foram utilizados.

A principal razão foi o facto de os gestores quererem que cada caminho de cada cliente/visitante fosse visto como um caminho diferente.

Podemos verificar pelo exemplo 2 que o mais utilizado dos quatro caminhos apresen-

tados seria, $< 2, 3, 4, 5, 6 >$, em vez do que seria de esperar como resultado dos outros algoritmos, ou seja, os caminhos $< 2, 3 >$ e $< 2, 3, 4 >$.

Example 2 Caminhos

Alguns caminhos possíveis de encontrar

$$\begin{aligned} p1 \Rightarrow &\quad < 2, 3, 4, 5, 6 > \\ p2 \Rightarrow &\quad < 2, 3, 4, 7, 9, 10 > \\ p3 \Rightarrow &\quad < 2, 3, 10, 9, 8, 7 > \\ p4 \Rightarrow &\quad < 2, 3, 4, 5, 6 > \end{aligned} \tag{4.1}$$

Como, normalmente, o caminho de cada visita se encontra com "falhas", tornou-se imperativo encontra-las e colmatá-las, tal é conseguido através das funções *smoothing*, *change*, *removeRepetitions* e $transform$.

Smoothing - Através de um conjunto de âncoras que definem um caminho, o smoothing atua sobre as âncoras cujo *timestamp* da última deteção e o da primeira seja igual a uma janela temporal de 120 segundos. Com esse conjunto vai ser verificado se a primeira deteção deverá ser alterada. Para que isso suceda é necessário que o número de vezes que a âncora que em mais ocasiões ocorre na janela escolhida, apareça, pelo menos, 60% das vezes. Caso isto não aconteça o mesmo processo é efetuado numa amostra igual a metade do tempo da janela seleccionada (i.e., 60 segundos). Se o resultado for igual ao prévio é feita a mesma verificação numa janela menor (i.e. 30 segundos).

Ainda, caso nada do que é descrito ocorra a âncora inicial é mantida e a janela é deslizada uma âncora.

Algorithm 4.6.2: SMOOTHING($Path$)

```
procedure SMOOTHING(Path)
  i ← 0
  while i < len(Path)
    do  y ← FIND(120, Path, i)
        if (y >= 460)
          then Path[i] ← y
          else y ← FIND(60, Path, i)
               if (y >= 60)
                 then Path[i] ← y
                 else y ← FIND(30, Path, i)
                      if (y >= 60)
                        then Path[i] ← y
        i ← i + 1
  return (Path)

procedure FIND(Time, Path, i)
  t ← 0
  k ← i
  while t < Time
    do  if i <= len(Path) − 1
          then a_0 ← Path[k].ancor
               t_0 ← Path[k].timestamp
               if k <= len(Path) − 2
                 then a_1 ← Path[k + 1].ancor
                      t_1 ← Path[k + 1].timestamp
                      t ← t_1 − t_0
                 else t ← Time
               if Time = 30
                 then t_s.append ← t_0
               ancor.append ← a_0
  counting ← Count every ocurrence of equal ancors
  r ← Verify which ocurrence is higher in counting
  return (r)
```

Change - Ao percorrer o caminho (constituído pelas âncoras em que o dispositivo foi detetado), por vezes, não é possível chegar, diretamente, de uma ancora à seguinte, assim, se tivermos a ancora 3 seguida da ancora 5, sabemos que o dispositivo teve de passar pela ancora 4. Quando este tipo de problemas acontece, é acrescentado o caminho mais curto para chegar de uma âncora à outra, no caso do exemplo 3 apresentado, ficaríamos com o caminho da equação 4.2.

O algoritmo utilizado para conseguir chegar às âncoras em falta foi executado apenas uma vez com o propósito de conseguir uma variável com todas as possibilidades de passagem entre os nós do caminho. Este algoritmo que permite encontrar o caminho mais curto entre dois nós utilizou a técnica de *backtracking*, ou seja, tenta cada possibilidade na sua vez, até encontrar a solução.

removeRepetitions - Esta função permite remover algum tipo de repetições, no caso de encontrar-mos a mesma âncora repetida em tempos sequenciais, ficamos apenas com essa ancora uma vez e se estivermos na presença de um padrão (de no máximo duas ancoras), ficamos com a primeira âncora e a última, como podemos ver no exemplo 3 equação 4.3.

Uma das verificações efetuadas, devido ao facto de algumas das âncoras, por vezes, falharem deteções, foi se existe, ou não, mudança de piso. Verificamos se quando há alteração para o mesmo piso, a âncora detetada é a inicial. Caso seja a mesma âncora, todas as âncoras intermédias serão removidas, o mesmo é efetuado conforme alteração do *recieved signal strength* (rss). Se o rss for considerado forte (abaixo dos 75), enquanto não for melhor, avança-se nas âncoras até tornar a ser forte, quando o rss voltar a ser forte e, se a âncora inicial for igual à final, as âncoras intermédias são removidas.

transform - Após a limpeza mencionada estar feita, transformamos as âncoras em zonas. como vamos encontrar repetições equivalentes às anteriores, as zonas passam por uma limpeza de repetições antes de serem utilizadas na criação dos caminhos (equação 4.4)

Example 3 Limpeza de um caminho

Se tivermos o caminho

$$p =< 1, 2, 2, 2, 3, 5, 6, 7, 6, 7, 6, 7, 6, 7, 9 >$$

o resultado será:

$$\begin{aligned} c &= change(p) \\ c &\Rightarrow < 1, 2, 2, 2, 3, 4, 5, 6, 7, 6, 7, 6, 7, 6, 7, 8, 9 > \end{aligned} \tag{4.2}$$

$$\begin{aligned} d &= removeRepetitions(c) \\ d &\Rightarrow < 1, 2, 3, 4, 5, 6, 7, 8, 9 > \end{aligned} \tag{4.3}$$

$$\begin{aligned} e &= transform(d) \\ e &\Rightarrow < 5, 5, 5, 6, 7, 7, 8, 8 > \\ e &\Rightarrow < 5, 6, 7, 8 > \end{aligned} \tag{4.4}$$

Capítulo 5

Modelação

Neste capítulo descrevemos os processos de análise de dados e modelação que permitem responder às perguntas colocadas pelos gestores. Para cada pergunta teremos um processo que foi implementado e integrado no BIPS. A escolha do processo de modelação mais adequado foi feita utilizando uma amostra (dados referentes a um mês).

Criámos e estudámos gráficos para verificar se existiam surpresas inesperadas, e saber a densidade do número de visitas, teve como propósito saber se existiam zonas mais importantes, pisos mais utilizados, lojas mais visitadas e acessos mais utilizados.

5.1 Qual o número de visitas por dia, hora, zona, piso ou acesso?

No que se refere aos dados relevantes para responder a esta questão, foram utilizados o número de dispositivos que passearam num dia no CC, o momento de entrada e de saída e as âncoras que detetaram esse dispositivo ao longo da visita. Todos os valores apresentados como resultado desta questão (e apresentados no produto final) são conseguidos através de técnicas de estatística descritiva, em especial através da média.

Visitas por Dia - Para sabermos o número de visitas por dia é necessário verificar, em primeiro lugar, se o dispositivo fez uma ou mais no mesmo dia. O número de visitas durante um determinado período de tempo é apresentado num gráfico de linhas. No

caso da amostra descrita, a figura 5.1 mostra-nos que, por regra, aos fins de semana o número de visitas aumenta. Podemos ainda estudar a distribuição do número de visitas diárias usando um histograma com uma curva de densidade, através das figuras 5.1 e 5.2, conseguimos ter a noção de que, na maioria dos dias, o número de visitas ao CC está entre as 2500 e 3000.

Figura 5.1: Visitas por dia no mês de Junho

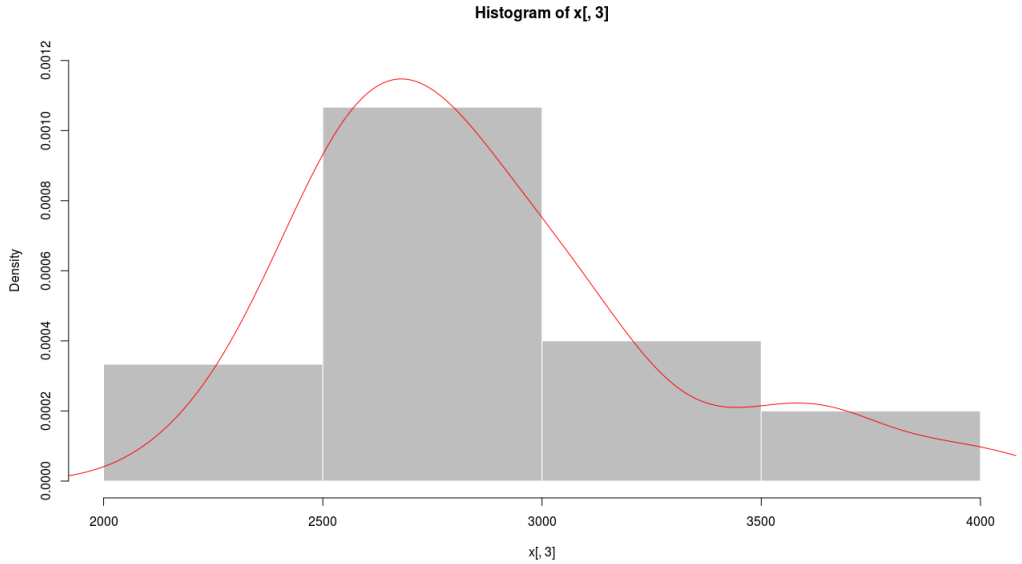

Figura 5.2: Densidade das Visitas no mês de Junho. No eixo dos x encontramos o número de visitas.

Visitas por Hora

Para se analisar o número de visitas por hora, dividimos o período de funcionamento diário em intervalos de uma hora. Os intervalos são respresentados pela sua hora de início. Num gráfico de linhas como o da figura 5.3, cada linha representa a evolução do número de visitas numa determinada hora ao longo dos dias. Tal como as visitas por dia utilizámos estatística descritiva para analisar o número de visitas por hora.

No nosso exemplo (figura 5.3) conseguimos observar que em média, encontram-se mais pessoas a visitar o CC pelas quatro da tarde e que o pico de visitas no mês, em quase todas as horas, foi no segundo fim de semana, de dia 8 a dia 10 de Junho.

Importa referir que o dia 10 de Junho, Segunda-Feira, foi feriado em Portugal, justificando o aumento de visitas nesse dia, principalmente, comparando com as restantes Segundas-Feiras. Por outro lado, a maior parte das horas teve uma descida considerável no Domingo 23 de Junho, esta diminuição pode ter uma relação direta com o facto de nos encontrarmos no final do mês, e paralelamente com o início da época de férias escolares.

Excetuando as horas de entrada e saída (10h - X10 e 23h - X23), a maior densidade

encontra-se entre os 400 e 600 visitantes. Podemos também perceber pela figura 5.4 que da parte da tarde, a partir das 14 horas, a probabilidade de observarmos um número acima das 800 visitas é alta, esta subida de visitantes começa a diminuir a partir das 18 horas.

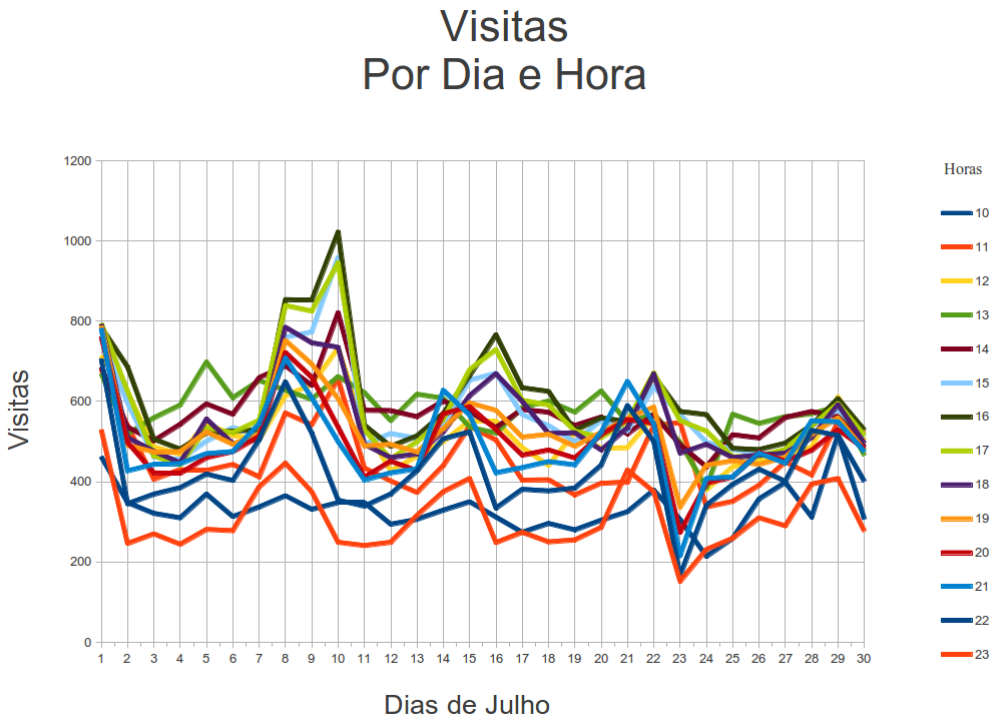

Figura 5.3: Visitas por Hora do dia no mês de Junho

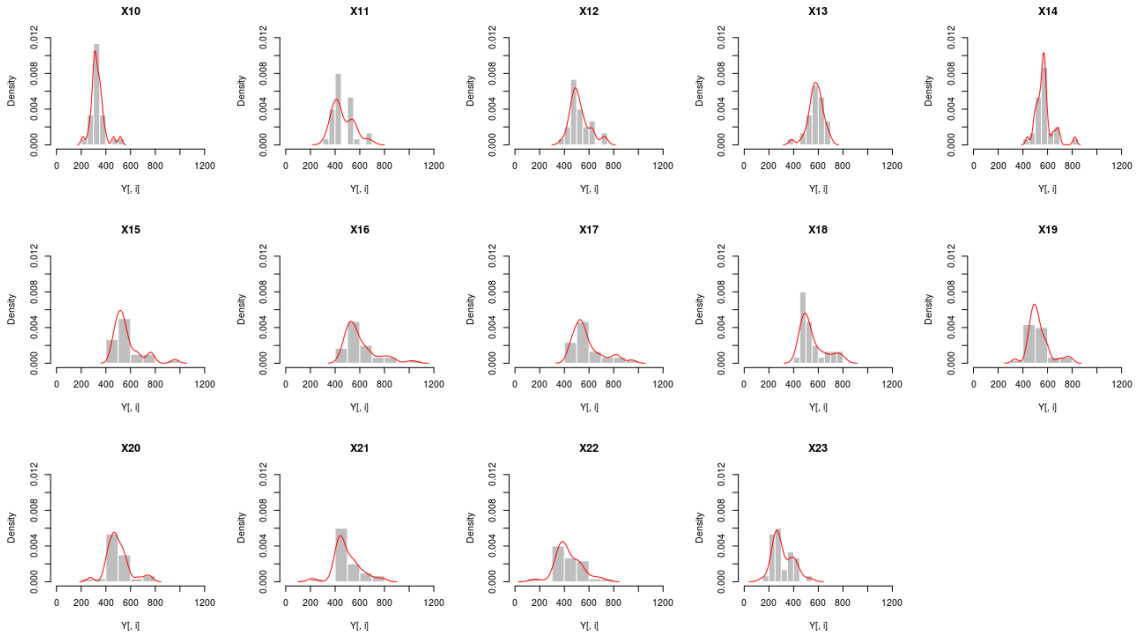

Figura 5.4: Densidade das Visitas por Hora no mês de Junho [ver apêndice A]

Visitas por Zona

Depois de associarmos cada âncora a uma zona, foi utilizada estatística descritiva para se conseguir obter a contagem do número de visitas efetuadas em cada uma das zonas. Analisando a amostra obtida do CC em estudo, podemos observar, através do gráfico circular da figura 5.5, que as zonas por onde a maioria dos visitantes passam são as zonas $z7$ e a $z1$.

Sabemos, através da figura 5.6 que, no máximo encontramos 3000 visitantes por zona e que na maioria das vezes este número se encontra entre os 1000 e 2000 visitantes.

A resposta à questão "**Qual o número de visitantes a passar na praça da alimentação?**"para a amostra de dados aqui estudada é de 1933 visitantes, tal como podemos observar na figura 5.5[1].

[1]Este gráfico mostra-nos a percentagem de visitantes (em relação a todos os que apareceram no CC) que passaram por cada uma das zonas.

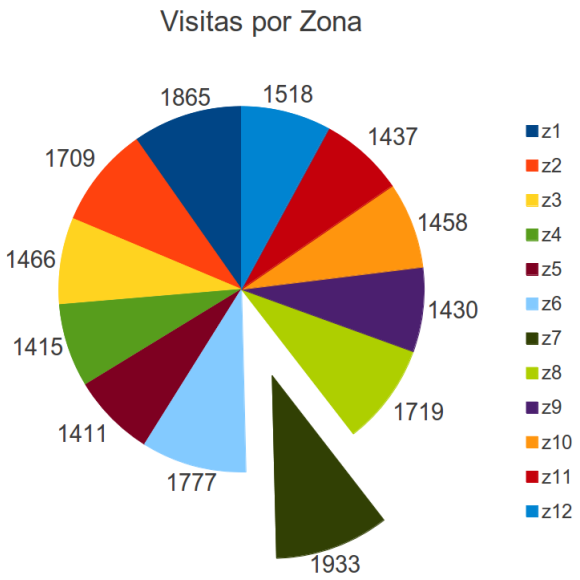

Figura 5.5: Visitas por Zona no mês de Junho

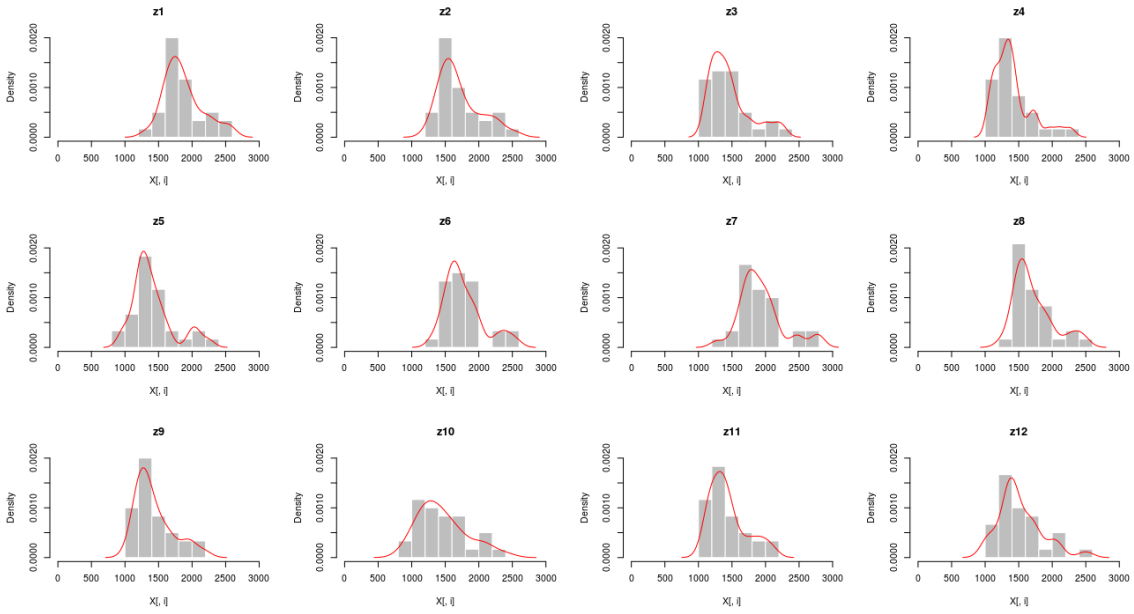

Figura 5.6: Densidade das Visitas por Zona no mês de Junho [ver apêndice A]

Visitas por Piso

Para os pisos necessitamos de ter as âncoras em que o dispositivo foi detectado. A partir delas conseguimos saber em que piso o visitante se encontra. O CC onde os dados foram adquiridos está dividido em dois pisos. Foram aplicadas tecnicas de estatística descritiva para alcançar o número de visitas por Piso. Através do gráfico da figura 5.8 sabemos que, no conjunto de dados recolhido do CC em estudo, o número de visitantes que passam pelos dois pisos é equivalente, apesar de ser um pouco maior no piso 1 (figura 5.7) factor que consideramos fácilmente justificável já que a maioria das portas de acesso se encontram no primeiro piso.

Conseguimos ver que no piso 1 a probabilidade do número de visitantes estar entre os 2250 e 2500 é alta, já no piso 2 o número encontra-se entre os 2000 e 2250. Como vemos a densidade de visitas nos dois pisos é muito proxima.

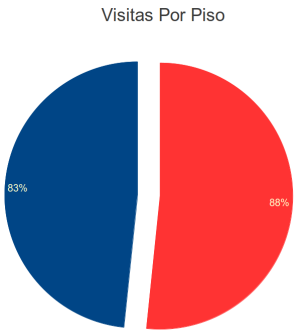

Figura 5.7: Visitas por Piso no mês de Junho

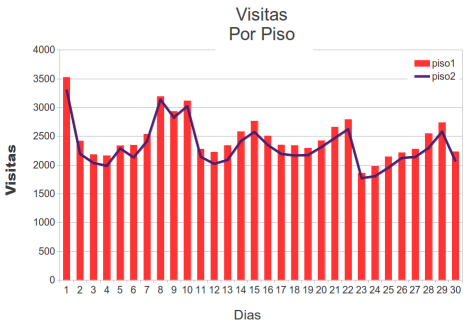

Figura 5.8: Comparação do número de visitas por Piso no mês de Junho

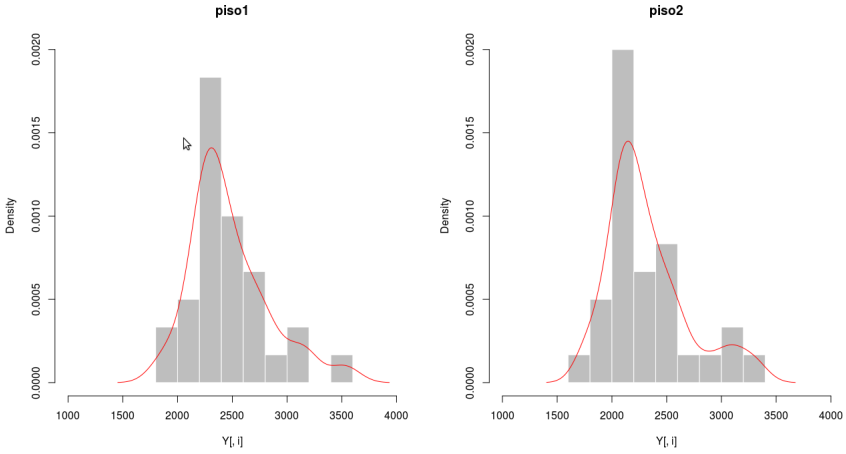

Figura 5.9: Densidade das Visitas por Piso no mês de Junho

Visitas por Acesso

Tal como já foi mencionado anteriormente, para conseguirmos saber qual o acesso por onde o visitante entrou, precisamos de ter a primeira âncora de detecção do visitante. Encontramos qual o Acesso associado a essa âncora e procedemos à divisão das visitas por Acesso.

Podemos ver, pela figura 5.11, que o acesso mais escolhido pelos visitantes no mês de Junho é o K, na mesma figura vemos que este acesso K se encontra quase sempre acima dos restantes. Apenas em três dias do mês o acesso G contem maior número que o acesso K.

Quanto ao acesso I a âncora referente a este acesso encontrou-se em baixo durante o mês a que se refere a recolha dos dados, por isso mesmo, apresenta um número de visitantes nulo.

Um aspeto interessante é o facto de, o acesso nem sempre ser o mesmo que a saída (figura 5.12). No caso de estarmos a falar do acesso J, é mais provável que a saída não seja a mesma, já se falarmos no acesso G ou K, a probabilidade é de o visitante sair pelo mesmo local. A figura 5.13 mostra-nos que a densidade mais alta encontra-se nos acessos D e F apesar de corresponder a um número de visitas mais baixo. Já para os acessos com o número de visitas mais alto, G e K, a densidade é consideravelmente mais baixa.

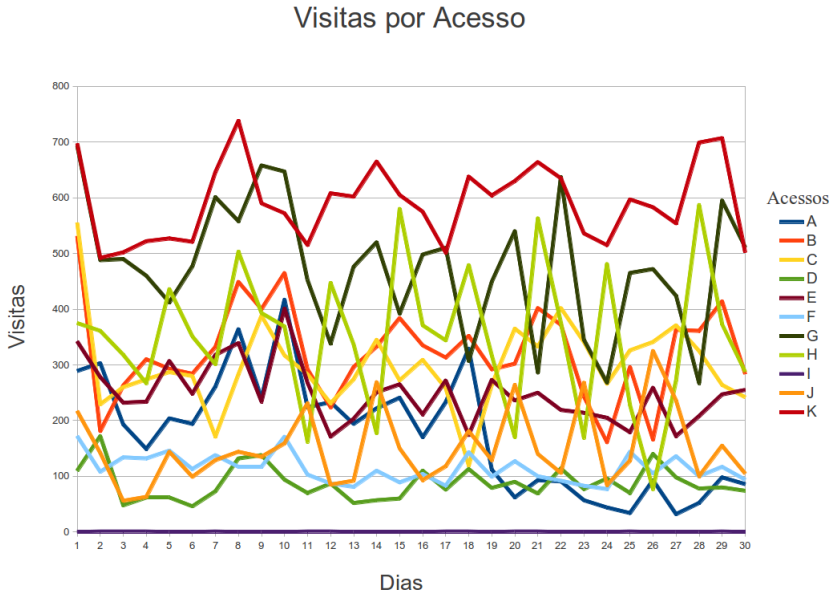

Figura 5.10: Visitas por Acesso no dia do mês de Junho

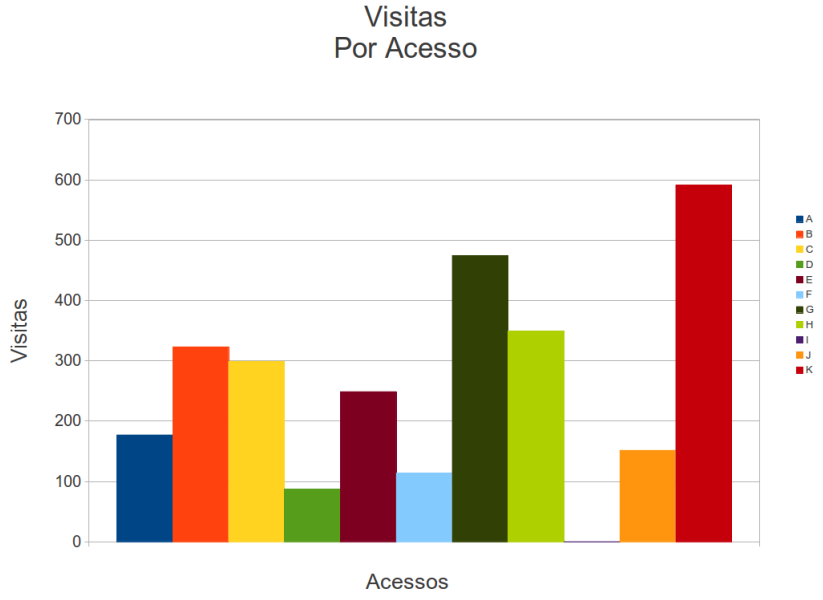

Figura 5.11: Visitas por Acesso no mês de Junho

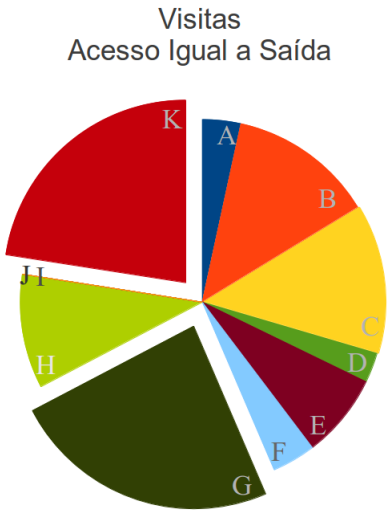

Figura 5.12: Visitas por Acesso Igual a Saída no mês de Junho

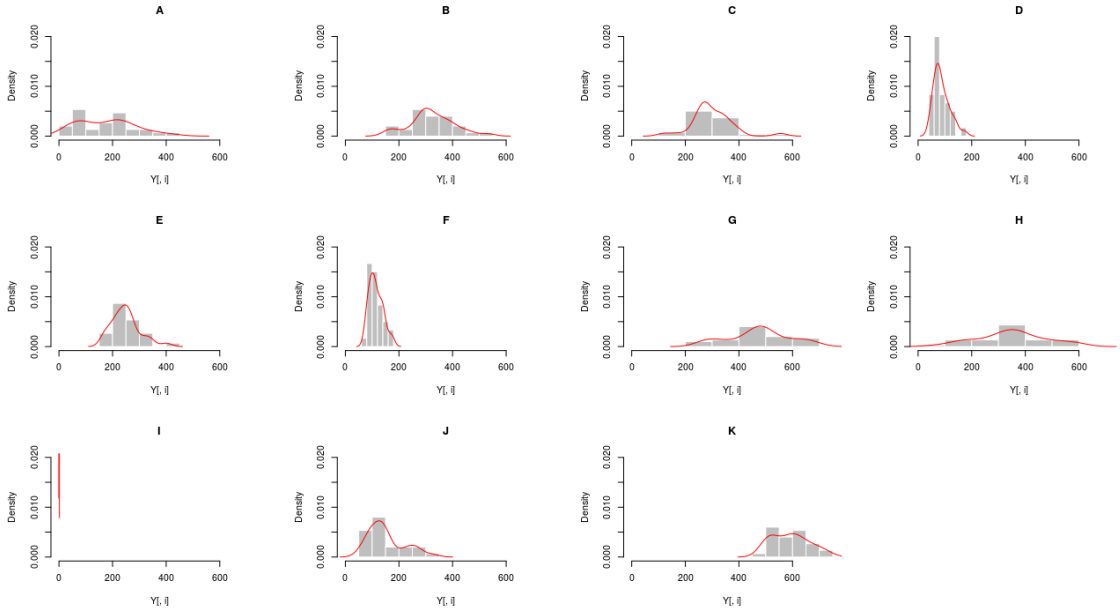

Figura 5.13: Densidade das Visitas por Acesso no mês de Junho [ver apêndice A]

5.2 Quais as cinco lojas mais visitadas?

Para poder responder a esta questão foi necessário ter as âncoras de deteção e momentos (*timestamps*) associados. Depois de se saber quais as lojas associadas às detecções conseguimos ordenar as lojas por número de visitas, permitindo obter as cinco mais visitadas. Através da figura 5.14 conseguimos ver as cinco lojas mais visitas. Esta são: L8,L17,L2,L11 e L13.

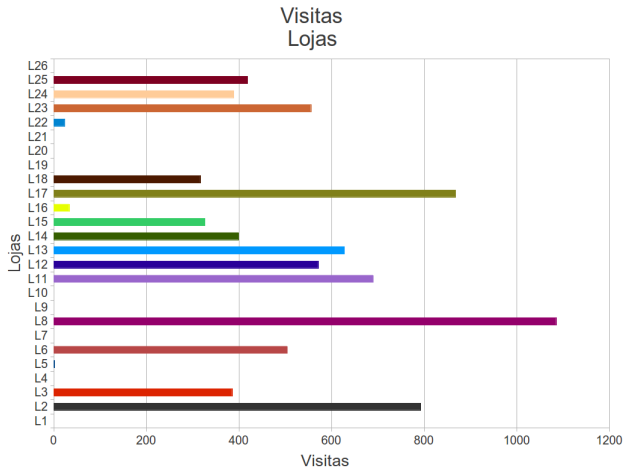

Figura 5.14: Visitas por Loja no mês de Junho

Conseguimos ter a noção de que uma boa parte das lojas tem normalmente entre os 500 e 1000 visitantes (figura 5.15) e que as restantes têm entre as 200 e 600 visitas. As lojas que se encontram a zero deve-se em grande parte ao facto das âncoras associadas a elas estarem em baixo (figura 5.16).

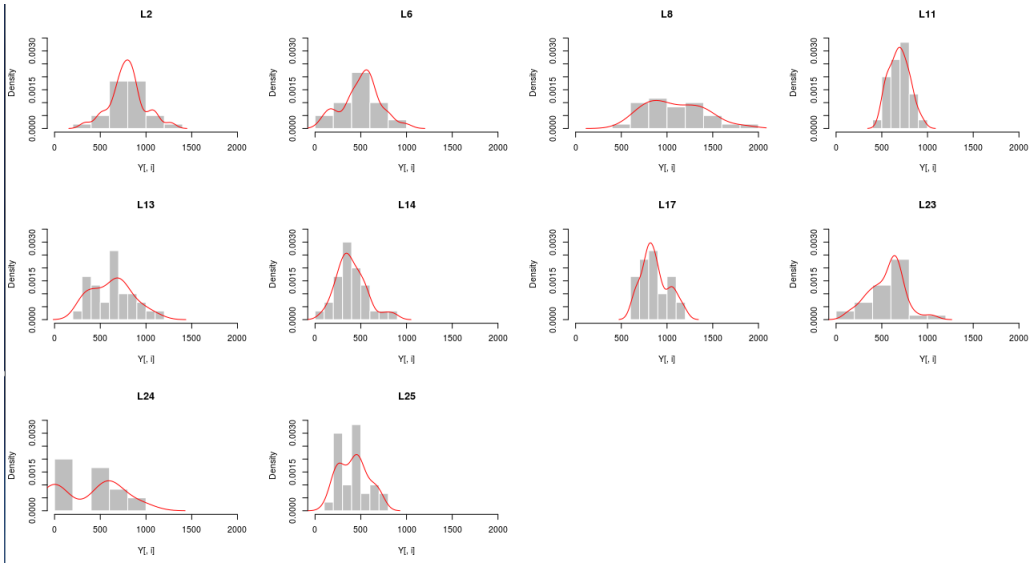

Figura 5.15: Densidade das Visitas por Loja no mês de Junho

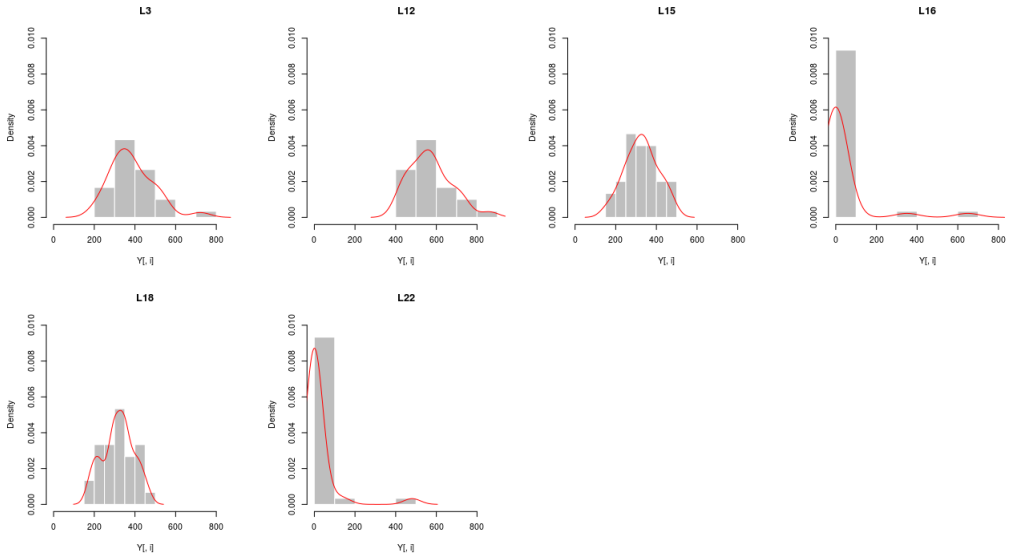

Figura 5.16: Densidade das Visitas por Loja no mês de Junho

5.3 Qual a loja escolhida como primeira paragem pela maioria dos clientes?

Da mesma forma que a questão anterior, foi necessário ter as âncoras de deteção e momentos (*timestamps*) associados. Sendo necessário no final ficar com a primeira detecção que seja interrompida por pelo menos 300 segundos e não mais de 3 horas. No que se refere à amostra em estudo, as lojas que os visitantes mais escolhem como primeira paragem são também as lojas que têm mais visitas ao longo do mês (apesar de ordenadas de outra forma), as cinco lojas mais visitadas como primeira paragem (L17,L8,L11,L13 e L2) não são, no entanto, as que têm uma maior densidade de visitantes (L22,L15,L16,L3 e L18).

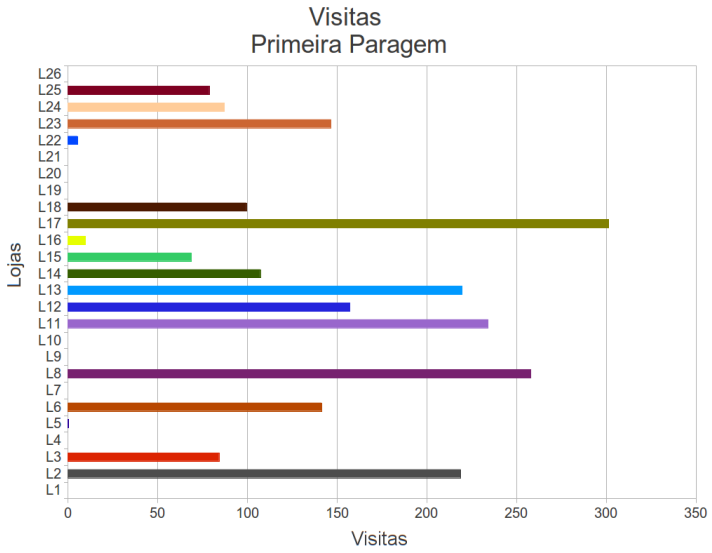

Figura 5.17: Primeira Paragem por Loja no mês de Junho

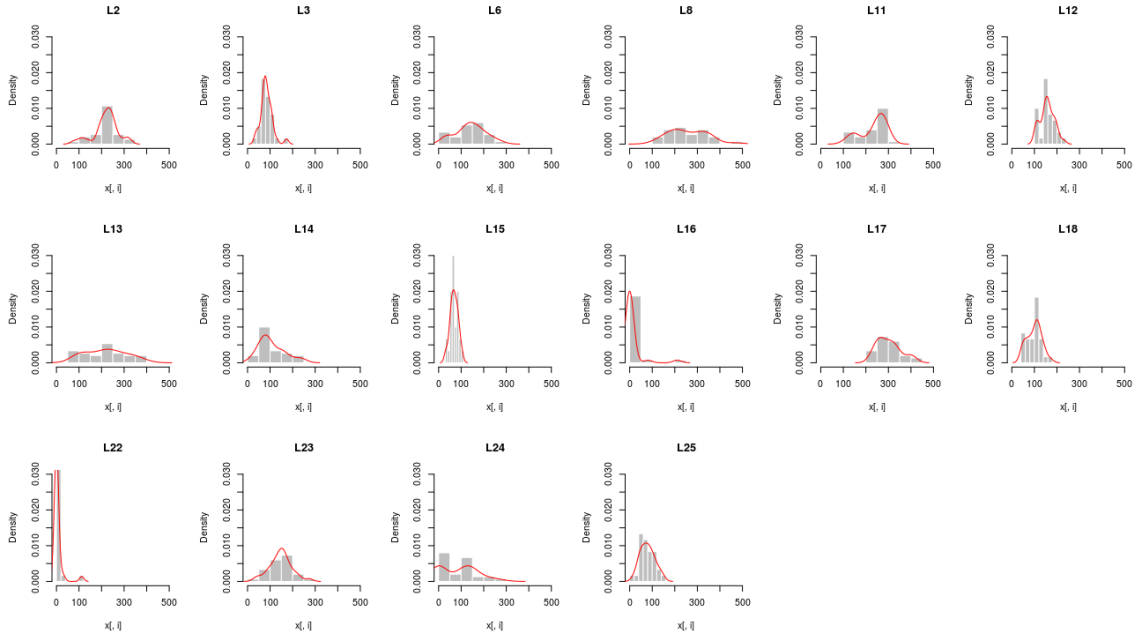

Figura 5.18: Densidade das Visitas na Primeira Paragem/Loja no mês de Junho [ver apêndice A]

5.4 Qual o tempo de uma visita, a fazer compras (*ShoppingTime*), de passeio (*DwellTime*) e passado na loja de primeira paragem?

Na resposta a esta questão foram também utilizadas técnicas de estatística descritiva. A partir dos dados já mencionados, conseguimos obter o tempo passado entre cada detecção. Através deste espaço de tempo, conseguimos saber se a visita ocorreu dentro de uma loja ou no corredor. Durante o mês de Junho, uma visita normalmente demora entre 5000 e 6200 segundos, a média do tempo gasto numa visita encontra-se entre os 5400 e 5600 segundos.

O diagrama de bigodes da figura 5.20 mostra-nos isso de uma forma simples e percetível e ainda nos permite ter a noção de que a maioria dos visitantes gasta entre os 5600 e 5800 segundos na sua visita. A figura 5.19 dá-nos a noção que a soma do *ShoppingTime* com o *DwellTime* é o tempo de visita. Percebemos que uma parte do

ShoppingTime é o tempo gasto na primeira paragem, esta parte é sempre acima dos 30% do *ShoppingTime*. Na maioria dos casos o tempo gasto na primeira paragem é superior ao gasto em cada uma das restantes lojas, quando nos referimos à amostra de dados em estudo.

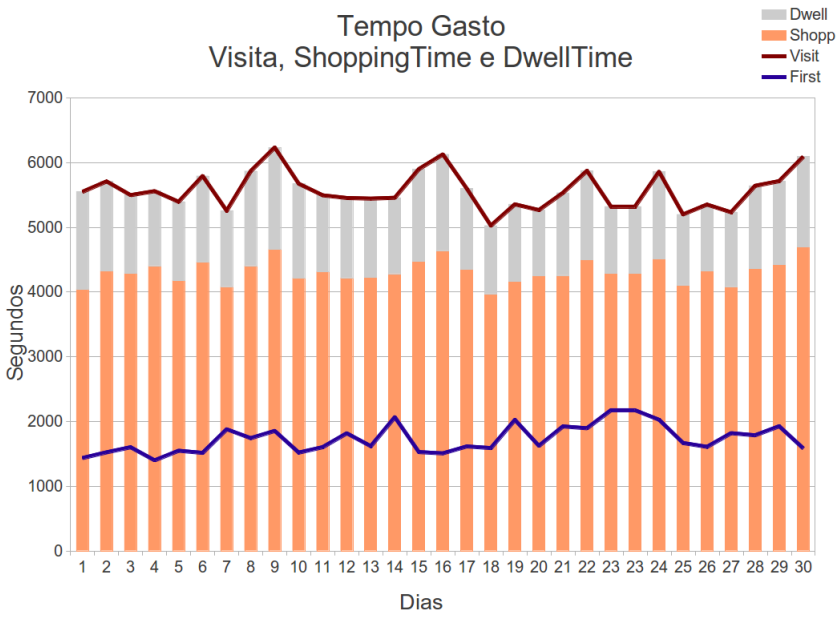

Figura 5.19: Comparação entre os tempos de visita, o tempo gasto pelos visitantes a fazer compras e o tempo que passam a passear no CC durante o mês de Junho

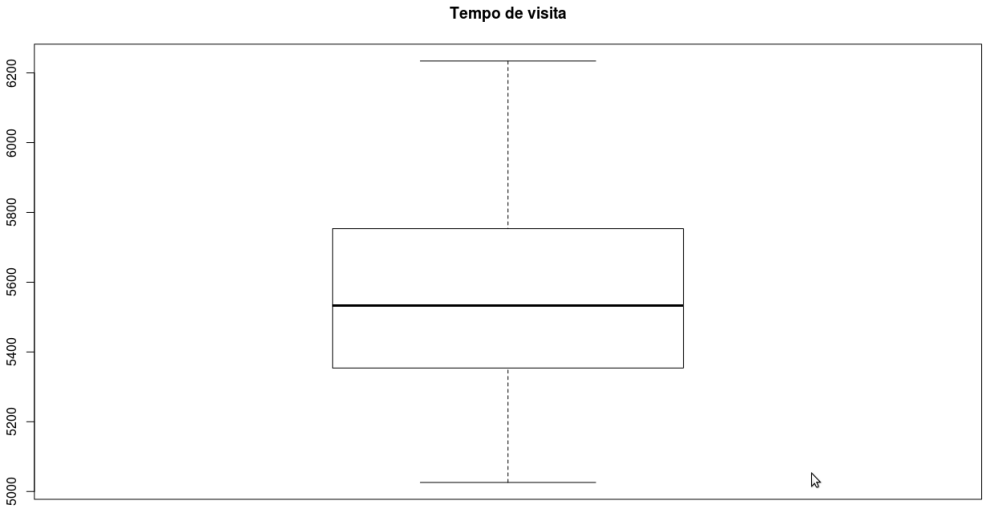

Figura 5.20: BoxPlot do Tempo de Visita no mês de Junho

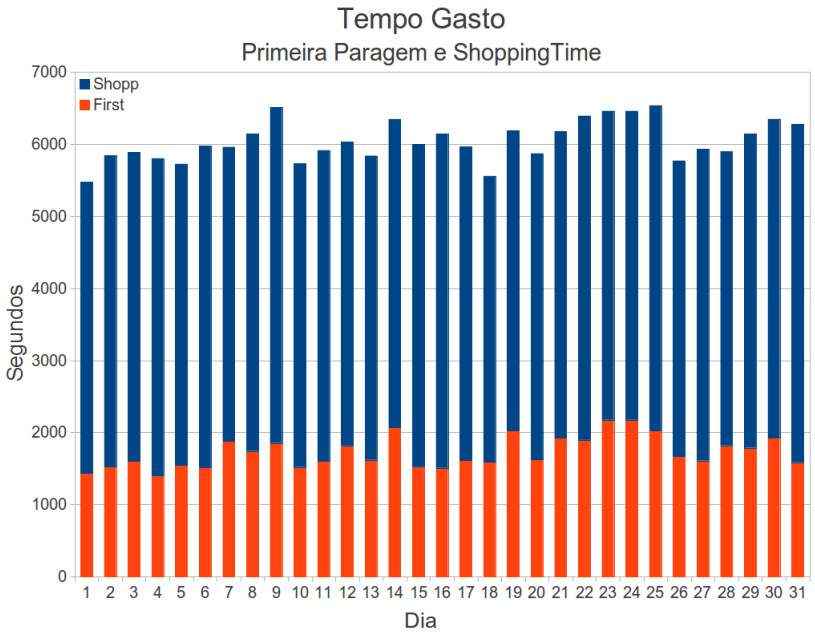

Figura 5.21: ShoppingTime e Primeira Paragem no mês de Junho

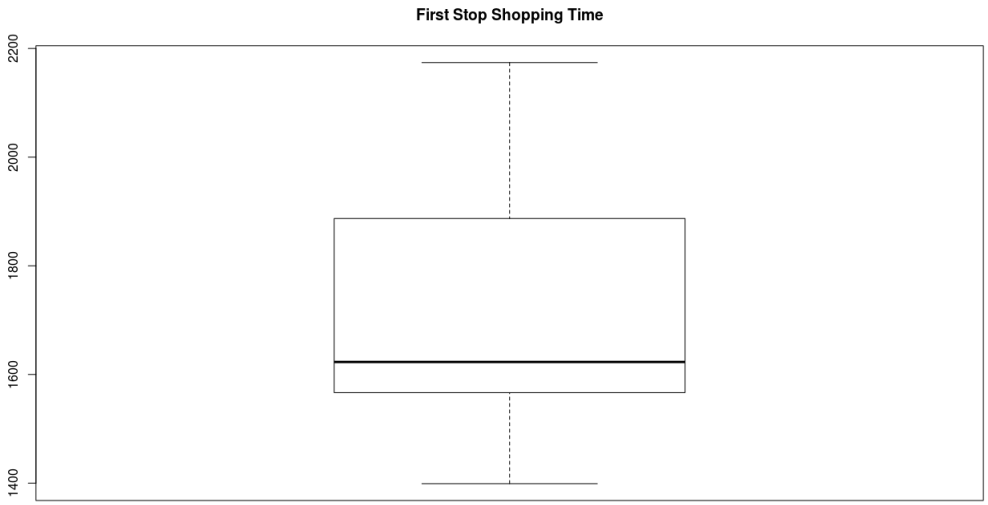

Figura 5.22: Densidade do Tempo gasto na Primeira Paragem no mês de Junho

5.5 Qual o tempo dispendido por Loja?

Tal como anteriormente, foram utilizadas técnicas de estatística descritiva. Os dados utilizados para a resposta a esta questão são os mesmo que foram mencionados na questão anterior. O resultado da aplicação da ferramenta, que permite responder a esta questão, à amostra em estudo é possível ser vizualizada através da figura 5.23, onde a Loja em que as pessoas passam mais tempo é a $L17$.

Apesar de tal ser verdade, podemos observar que na maioria dos dias, o tempo despendido nas visitas à loja L13 ultrapassam o tempo despendido à loja L17, três dias (1, 8 e 10 de Junho). Mesmo assim estes três valores não ultrapassam o maior tempo passado em L17 e por pouco passa a média do tempo gasto do mês nas lojas.

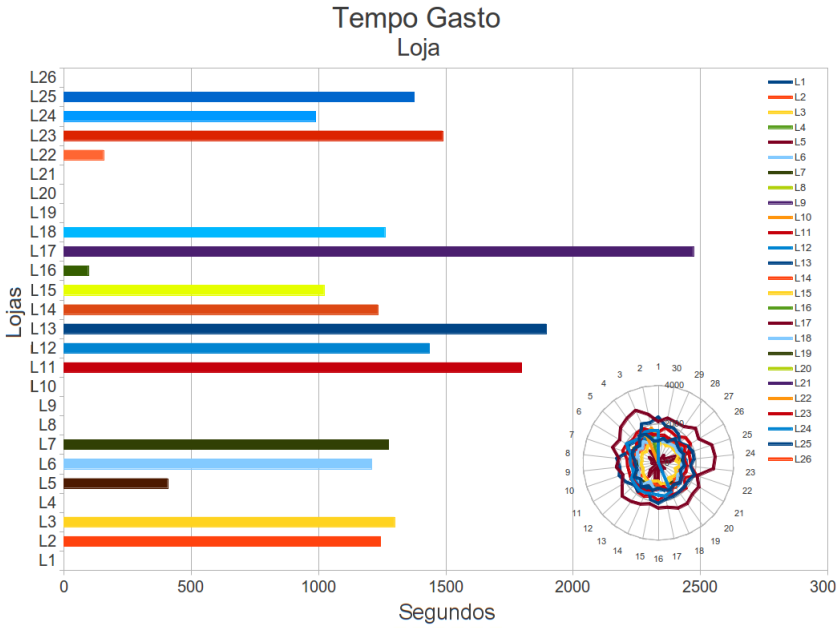

Figura 5.23: Tempo gasto por Loja no mês de Junho.

5.6 Qual o caminho mais utilizado pelos clientes a partir de um acesso?

Foram efetuados testes nas ferramentas de *data mining* mencionadas no capítulo 2. Tentámos obter resultados com o RapidMiner (figura 5.24) a partir de um conjunto de dados (figura 5.25) mas, este não nos permitiu obter resultados (figura 5.26), este erro deveu-se ao facto de existir um problema na ferramenta que será corrigido na próxima versão a ser lançada.

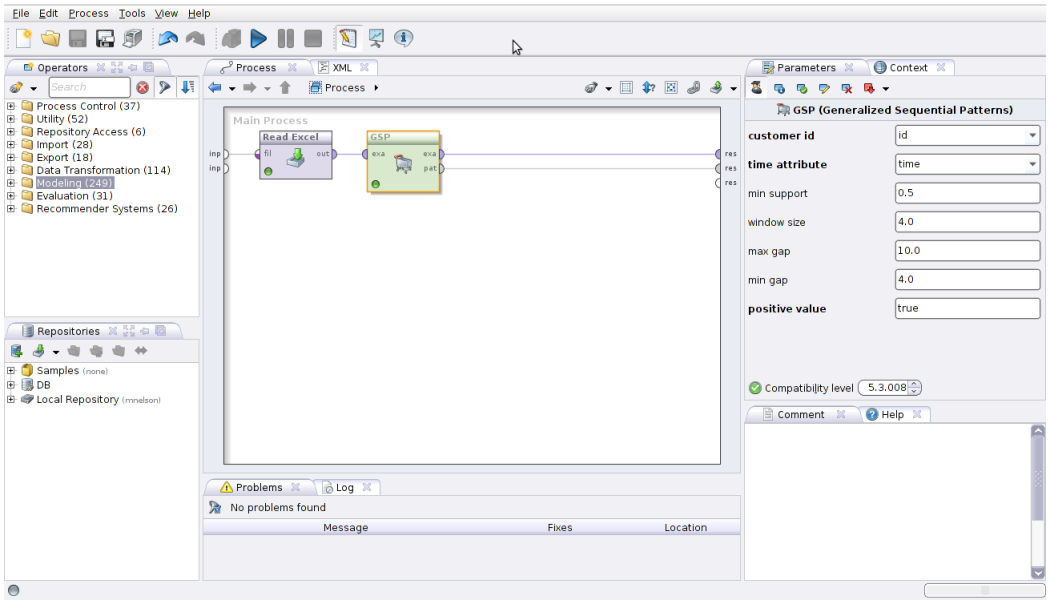

Figura 5.24: Utilização do GSP no RapidMiner

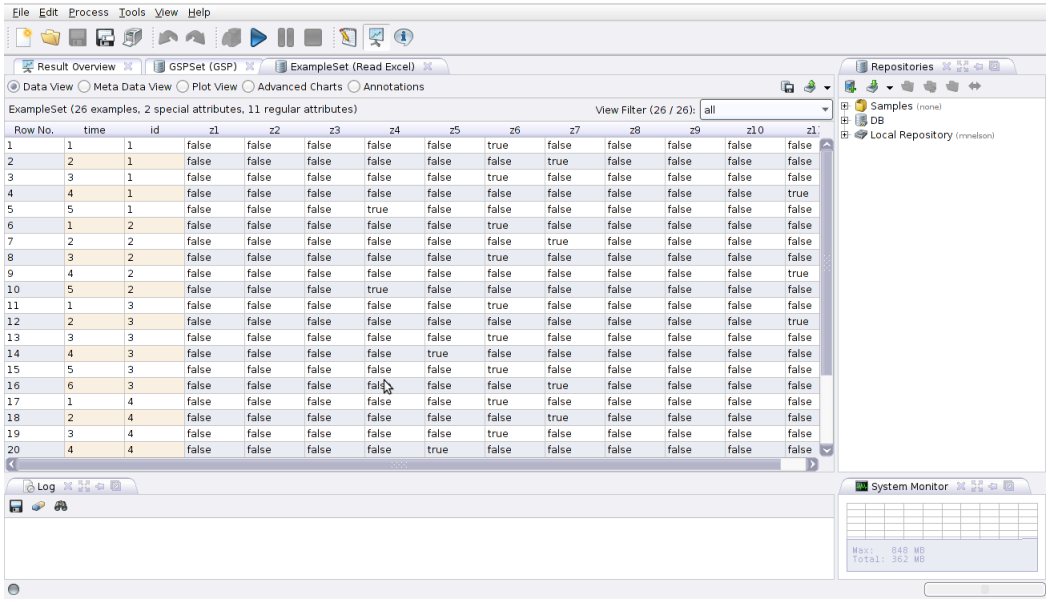

Figura 5.25: Conjunto de caminhos efetuados a partir do acesso A

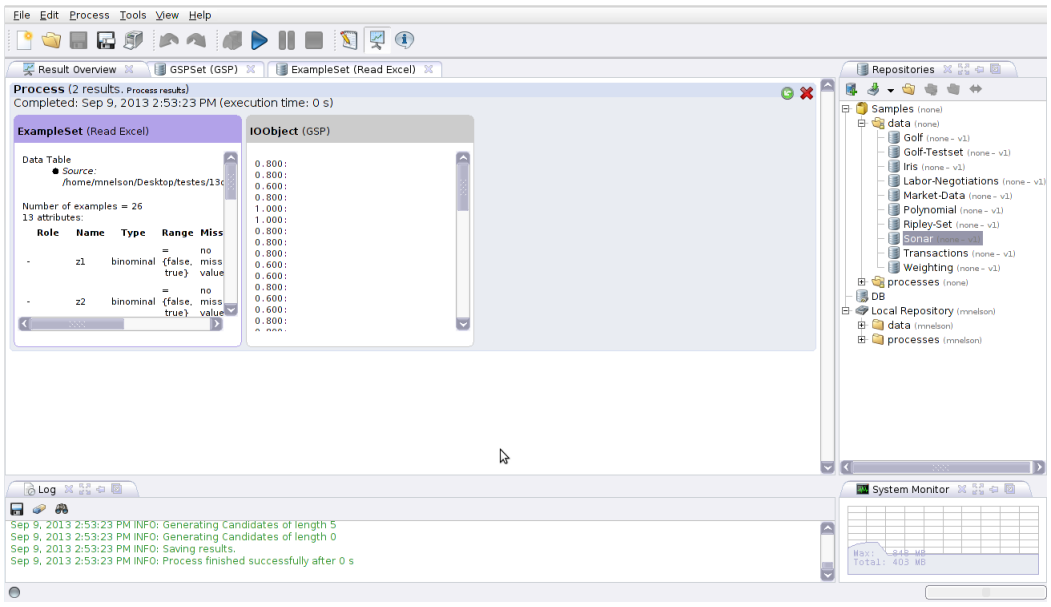

Figura 5.26: Resultado obtido com o GSP no RapidMiner

Quanto à ferramenta WEKA, esta foi utilizada para testar o mesmo algoritmo em detrimento do RapidMiner, o teste efetuado obteve o resultado apresentado nas figuras 5.27 e 5.28.

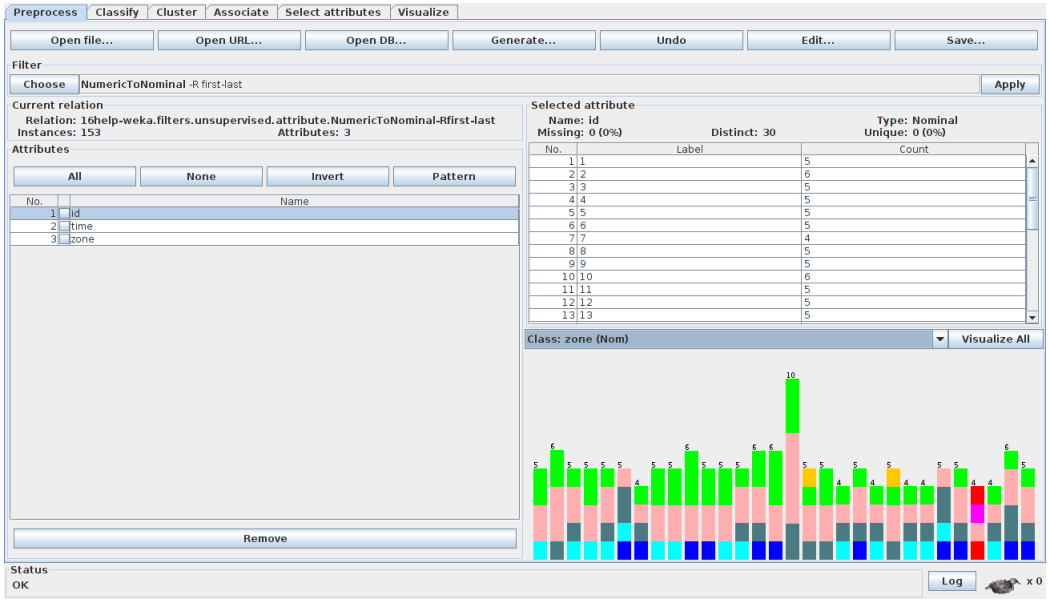

Figura 5.27: Teste efetuado com os dados do acesso A no Weka

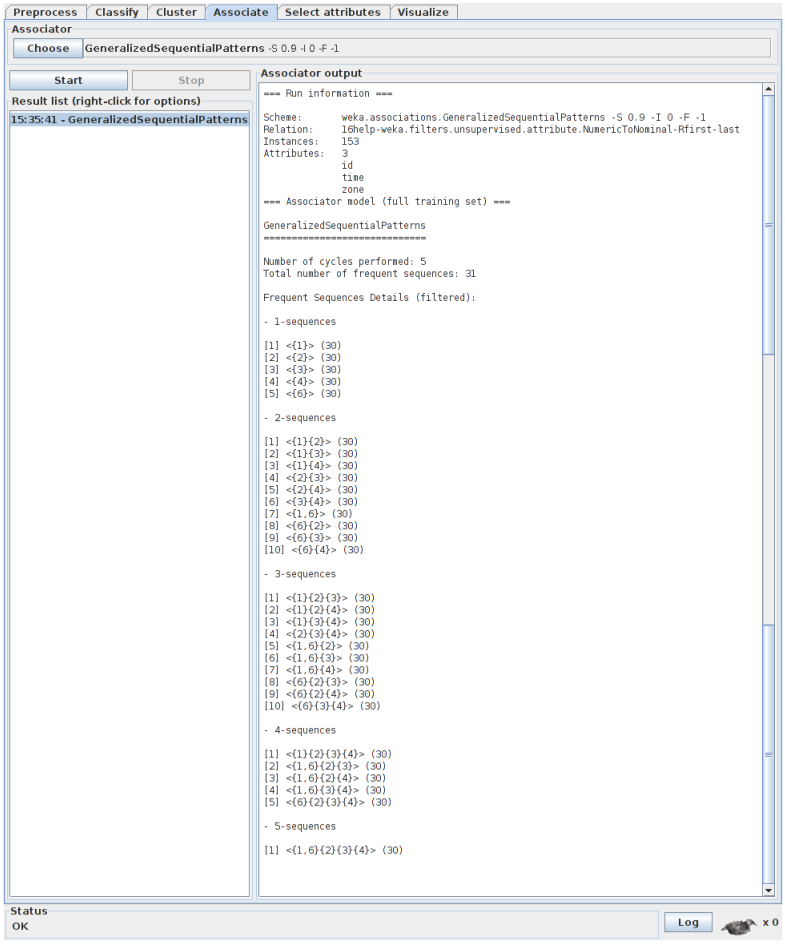

Figura 5.28: Resultados do teste com o GSP utilizando o Weka

No que se refere aos restantes algoritmos, SPAM e PrefixSPAN, foram testados com a ferramenta SPMF, em primeiro lugar foram utilizados conjuntos de poucos caminhos. Os resultados obtidos pelos dois algoritmos num desses testes está apresentados nas figuras 5.29 e 5.30. Depois de correr estes testes, o SPAM demora mais tempo e necessita de mais memória. Em seguida procedeu-se a testes com os dados dos caminhos referentes a um acesso que apresentamos na figura 5.31. Conseguimos resultados equivalentes a partir dos dois algoritmos, apesar de ter conseguido resultados rapidamente com o PrefixSpan (figura 5.32), com o SPAM esses resultados não foram atingidos devido às razões já referidas.

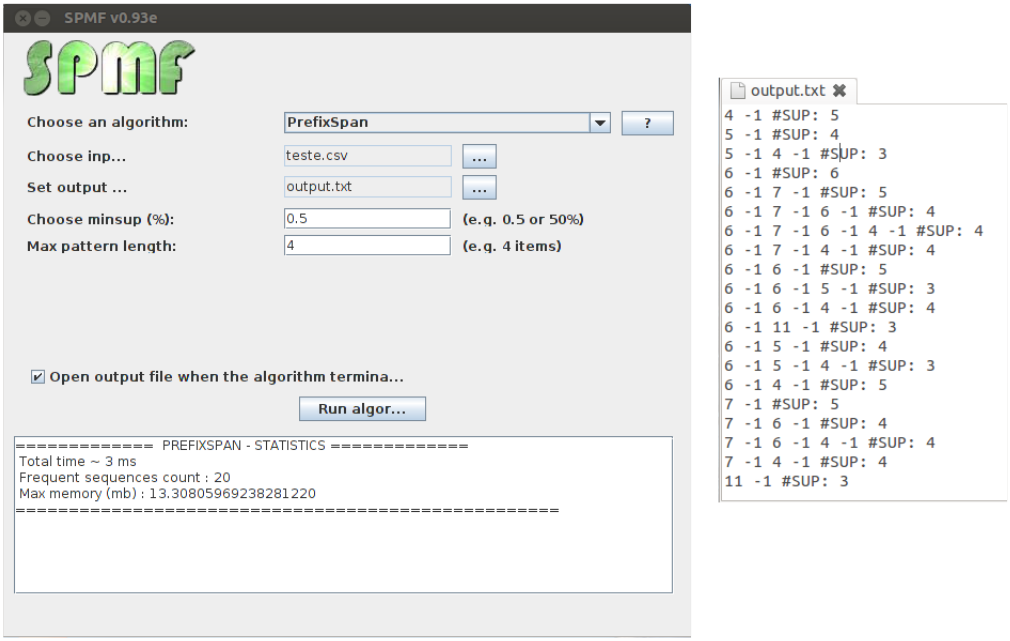

Figura 5.29: Resultados obtidos da utilização do PrefixSpan no SPMF em ficheiros de teste

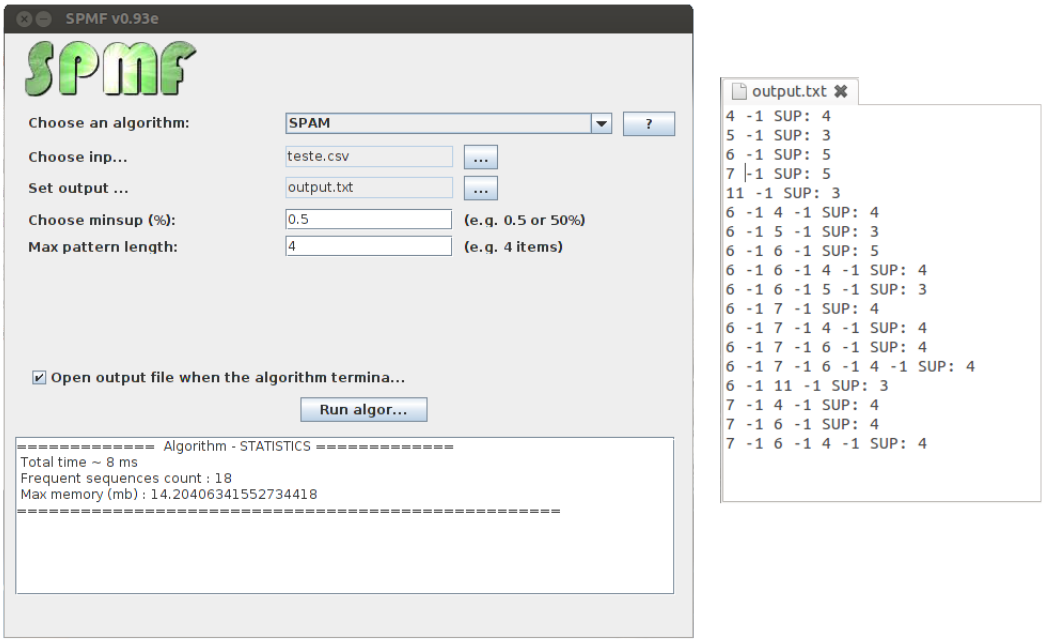

Figura 5.30: Resultados obtidos da utilização do SPAM no SPMF em ficheiros de teste

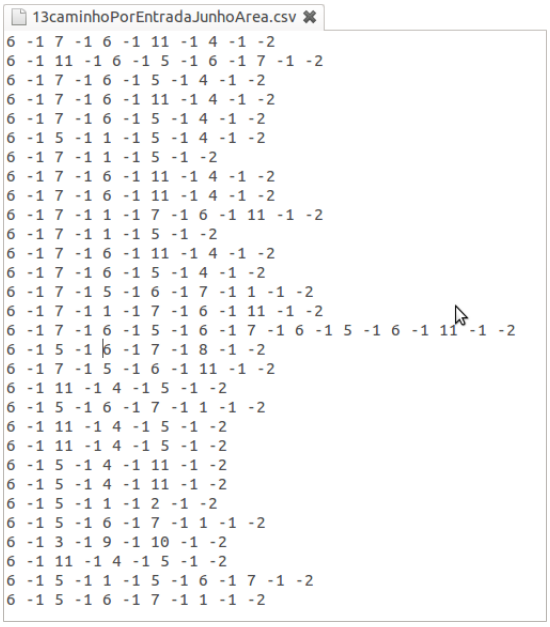

Figura 5.31: Conjunto de alguns caminhos efetuados a partir do acesso A, formatados para spmf

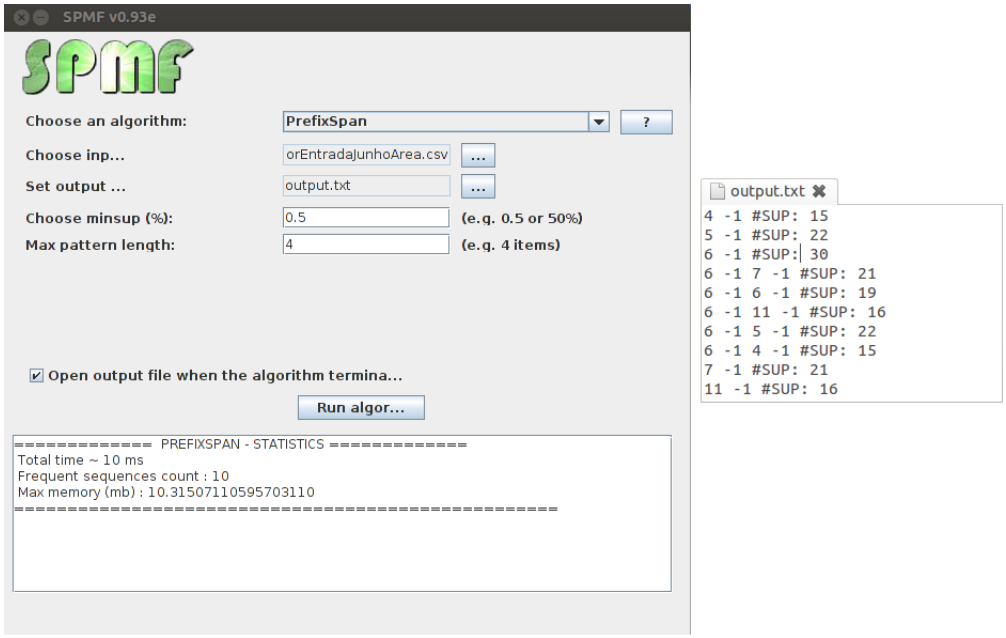

Figura 5.32: Resultados obtidos da utilização do PrefixSpan no SPMF

Para conseguirmos encontrar qual o caminho mais utilizado pelos visitantes foi criado um algoritmo [5.6.1] que utiliza a função de pré-processamento [algoritmo 4.6.1]. Para além do pré-processamento (descrito em capítulo anterior), utiliza a função $mostUsedPaths$. Esta função, em primeiro lugar corta os caminhos pelo número de zonas mínimas (min_Zones) escolhido, ficando com caminhos que contenham apenas um número mínimo de zonas diferentes. Em seguida é encontrado o caminho que aparece mais vezes. O facto dos caminhos ficarem definidos por zonas permite que, se existirem âncoras a falhar, não haja problemas com a criação dos caminhos possíveis.

Algorithm 5.6.1: PATHS($Data, AccessName, min_Zones$)

```
procedure PATHS(Data, AccessName, min_Zones)
  for each path ∈ Data
    do { AllPaths ← PREPROCESSPATHS(path, AccessName)
         returnedPath ← MOSTUSEDPATH(AllPaths, min_Zones)
  return (returnedPath)
```

Através do algoritmo descrito foi possível obter os grafos apresentados na figura 5.33 e através desta, concluímos que existem caminhos que, apesar de pertencerem a acessos diferentes começam na mesma zona (H e J ou A e B), já que existem zonas com mais que um acesso.

As zonas 7 e 8 aparecem em mais de metade dos caminhos, demonstrando serem as zonas mais visitadas.

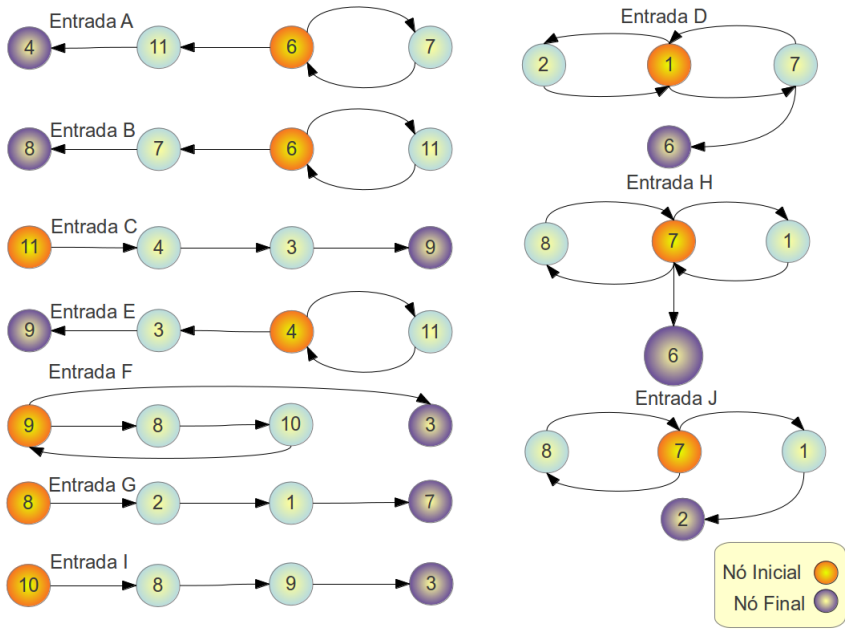

Figura 5.33: Caminhos mais utilizados por Acesso no mês de Junho

No sub-capítulo 2.4 mencionamos algoritmos de deteção de padrões, alguns dos resultados obtidos podem ser visualizados nas figuras 5.34, 5.35 e 5.36. Podemos ver que os resultados obtidos pelo Weka não nos trazem nada de novo e que o PrefixSpan e o SPAM, através do SPMF nos devolveram resultados agradáveis.

Estes caminhos, no produto final, são representados por um mapa do espaço com o caminho efetuado pelo visitantes, permitindo aos gestores compararem acessos.

5.7 Qual o caminho mais utilizado pelos clientes?

Na figura 5.34 podemos ver os caminhos mais e menos utilizados no mês de Junho pelos visitantes do CC e nas figuras 5.35 e 5.36 os caminhos cujo número de visitantes está acima da média de passagens e abaixo, respetivamente. Olhando para a figura 5.33 conseguimos perceber que os melhores caminhos por acesso passam, na sua maioria, por caminhos cujo número de visitantes se encontra acima da média.

Observamos que o caminho referente ao acesso G passa exatamente pelos caminhos com maior número de visitantes, os caminhos que passam por ligações cujo número

de visitantes está abaixo da média, encontram-se, mesmo assim, dentro das ligações com maior valor. O único caminho em que tal não acontece é referente ao acesso I que contém a ligação $10 - 8$, o número de passagens desta ligação encontra-se abaixo das 600.

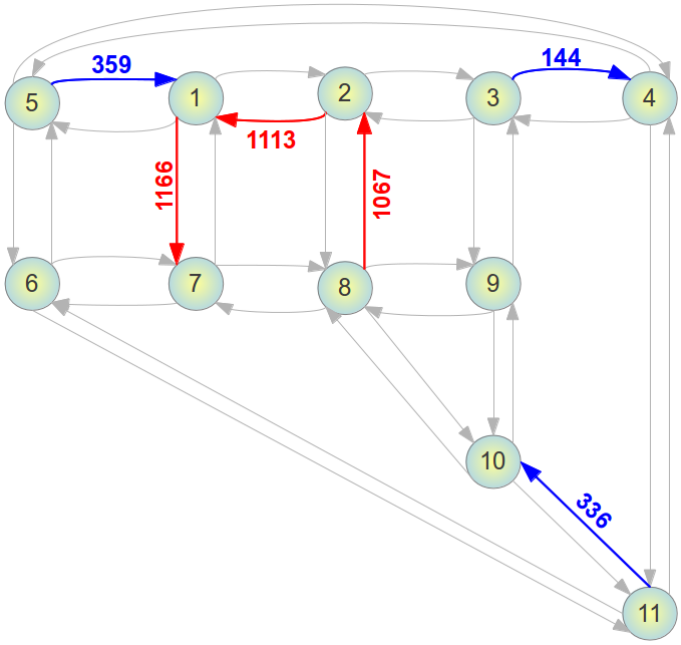

Figura 5.34: Caminhos mais/menos utilizados no mês de Junho

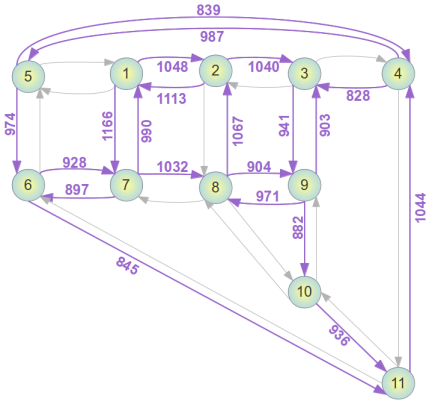

Figura 5.35: Caminhos com utilização acima da média no mês de Junho

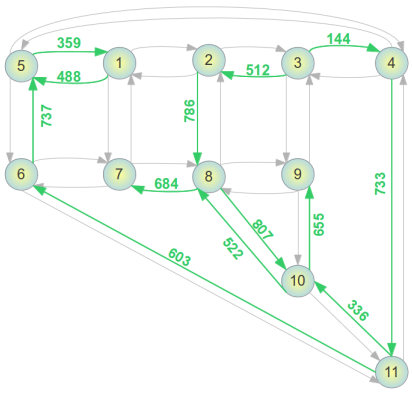

Figura 5.36: Caminhos com utilização abaixo da média no mês de Junho

5.8 Avaliação por parte do Cliente

Após verificação dos resultados obtidos na amostra de dados, os gestores do CC decidiram não incluir um pequeno conjunto de ferramentas criadas. Este conjunto é constituído pelas ferramentas que permitem responder às questões: "Qual a loja escolhida como primeira paragem pela maioria dos clientes?", "Qual o tempo passado na loja de primeira paragem?"e "Qual o tempo dispendido por Loja?". Estas ferramentas estão prontas e aptas a serem utilizadas no caso de outros clientes as acharem uma mais valia.

Capítulo 6

Produto Final

Neste capítulo descrevemos a implementação das técnicas e algoritmos descritos no capítulo anterior e a sua integração no BIPS.

A figura 6.1 descreve os vários passos de tratamento dos dados até à apresentação dos resultados. As deteções feitas pelas âncoras são enviadas para um servidor local e armazenadas na BD (C_1). Teremos acesso a esta BD para trabalhar os dados - através do conjunto de ferramentas desenvolvido atuando na parte de tratamento de dados - e obter resultados para as questões colocadas pelos gestores do CC.

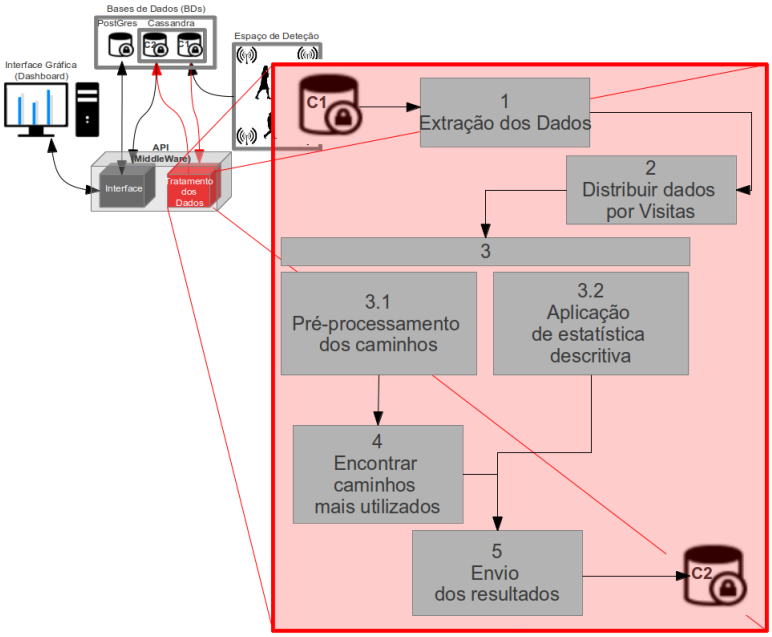

Figura 6.1: Conjunto de ferramentas

A descrição do conjunto de ferramentas criado [ver figura 6.1], mostra-nos os passos por que foi necessário passar e ultrapassar para conseguir a integração no BIPS.

1. **Ligação à BD** C_1 - Proceder à ligação à BD através da biblioteca de Python escolhida para o efeito.
2. **Extração dos dados** - Extração dos dados em blocos e guardá-los para serem tratados.
3. **Distribuir dados por visitas** - Tratar dos dados para se tornar mais rápido acesso aos mesmos. Foram divididos por visitas tendo o *timestamp* inicial e final da visita, a data em que foi feita, as paragens que o dispositivo fez e o caminho que percorreu.
4. **Tratamento dos dados**

   Pré-processamento de caminhos - Aplicar a função de pré-processamento [ver algoritmo 4.6.1].

   Aplicação de estatística descritiva - Aplicar técnicas de estatística descritiva [ver capítulo 5].
5. **Encontrar caminhos mais utilizados** - Aplicar a função *mostUsedPath* [ver algoritmo 5.6.1] para conseguir o caminho por acesso.
6. **Ligação à BD** C_2 - Proceder à ligação à BD
7. **Envio dos resultados** - Enviar os resultados de 4 para a BD C_2

Este conjunto de ferramentas é executado uma vez por dia, e os resultados enviados para uma BD C_2 [ver figura 6.1], aqui os dados inseridos já se encontram estruturados de uma forma diferente, para que a sua consulta pela API seja mais simples, o utilizador interage com a IG, que por sua vez, interage com a API que acede à BD C_2 para adquirir a informação como resposta às questões que o utilizador quer ver respondidas, enviando-as para a IG.

A todos os valores finais apresentados, acresce um factor, este foi decidido tendo em conta estudos efetuados sobre o número de pessoas que têm dispositivos móveis, o número de dispositivos móveis que cada pessoa tem em sua posse e se estes dispositivos têm, ou não, mecanismos de radiofrequência ligados.

Todas as pesquisas do BIPS permitem a escolha do intervalo de tempo que queremos consultar, esta escolha vai desde um dia, uma semana até um mês. Em seguida apresentamos a interface gráfica (IG) do BIPS.

6.1 *DashBoard*

O BipsBoard é a IG do BIPS e disponibiliza várias abas ao utilizador. A primeira, *DashBoard*, permite ao gestor ter acesso à informação mais importante tal como o número de visitas atual ou as cinco lojas mais visitadas, entre outros. Já a aba *Shopping Center* encontra-se dividida em 3 partes, oned podemos ver desde os dados gerais do CC aos caminhos mais percorridos. As abas *Levels*, *Zones* e *Accesses* apresentam informação sobre os respectivos nomes das abas. Finalmente a *Stores* apresenta informação sobre as lojas existentes na superfície comercial. Aqui apresentaremos as abas e características do BIPSBoard depois de ter sido feita a extenção do BIPS com o conjunto de ferramentas criado.

DashBoard - Para se poder consultar todas as informações referentes ao BIPS a AK optou por criar o *Dashboard*, este não é mais que uma IG onde se encontram todas as funcionalidades que o BIPS pode fornecer aos seus utilizadores. O *Dashboard*, apresentado na figura 6.2, permite ao utilizador uma vista geral dos valores, como o número de visitas por dia no mês ou as lojas mais visitadas.

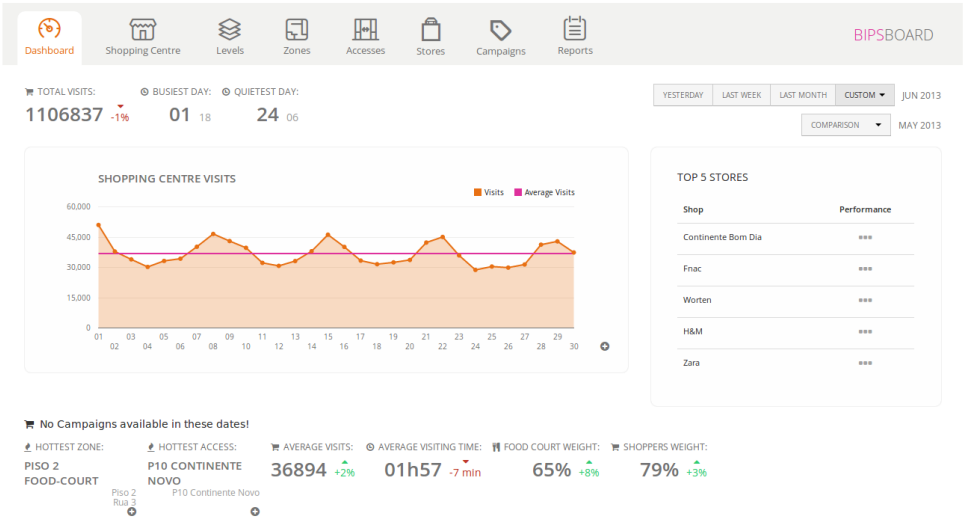

Figura 6.2: *Dashboard* para o mês de Junho

Shopping Center - A aba *Shopping Center* está dividida em três partes: *Metrics*, *Density* e *Paths*. Quando falamos de *Metrics* (figura 6.3), falamos de informação referente às visitas no CC, esta informação vai desde o número de visitas ao tempo passado a passear.

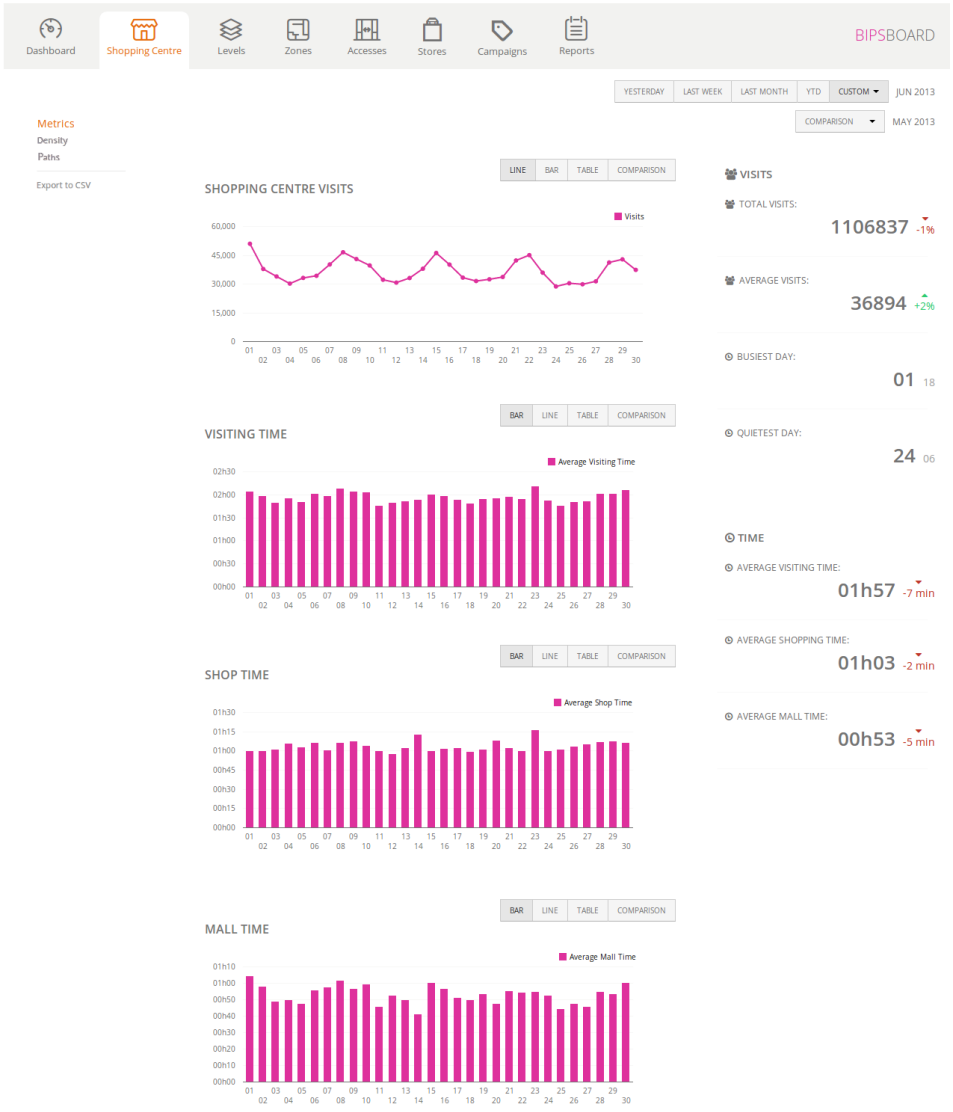

Figura 6.3: *Metrics* referentes às visitas para o mês de Junho (*Shopping Center*)

Os caminhos (figura 6.4) permitem ao utilizador conhecer quais os caminhos mais utilizados a partir de um acesso.

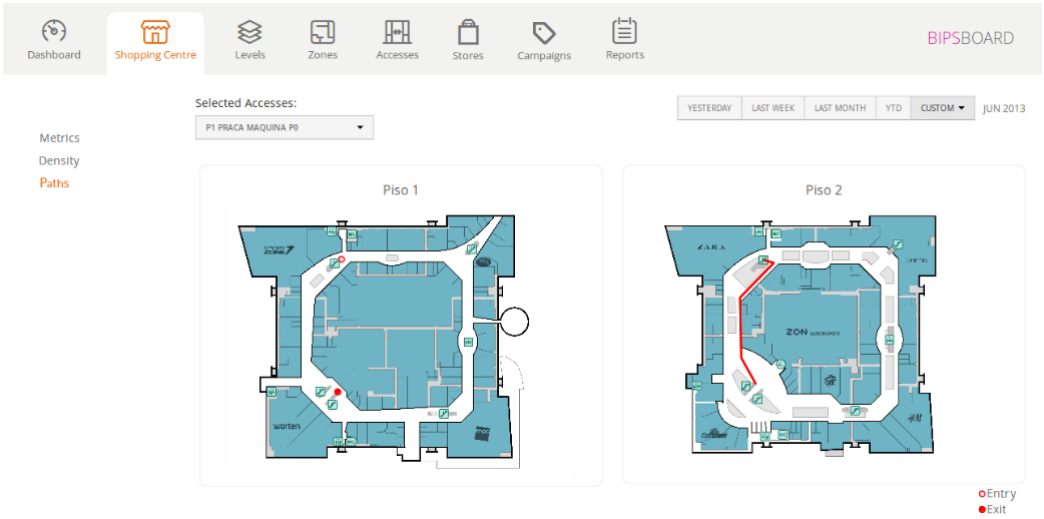

Figura 6.4: Caminhos mais utilizados a partir de um acesso no mês de Junho (*Shopping Center*)

Levels*, *Zones* e *Accesses - Nestas abas obtemos resultados quanto ao número de visitas e quanto ao tempo em que as mesmas decorrem. Na aba *Levels* (figura 6.5) mostramos o número de visitas por piso e o tempo despendido em cada uma. Na última não são incluídos tempos já que estes não foram pedidos pelo cliente.

Encontramos a mesma informação referente às zonas na aba *Zones* (figura 6.6) e os resultados referentes aos acesso na aba *Accesses* (figura 6.7).

Stores - Na aba *Dashboard* ficámos a saber quais as cinco lojas mais visitadas, na *Stores* temos um *ranking* de todas as lojas com o respetivo número de visitantes associado. Para além disto, escolhendo uma loja, conseguimos saber que lojas foram visitadas juntamente com a loja escolhida.

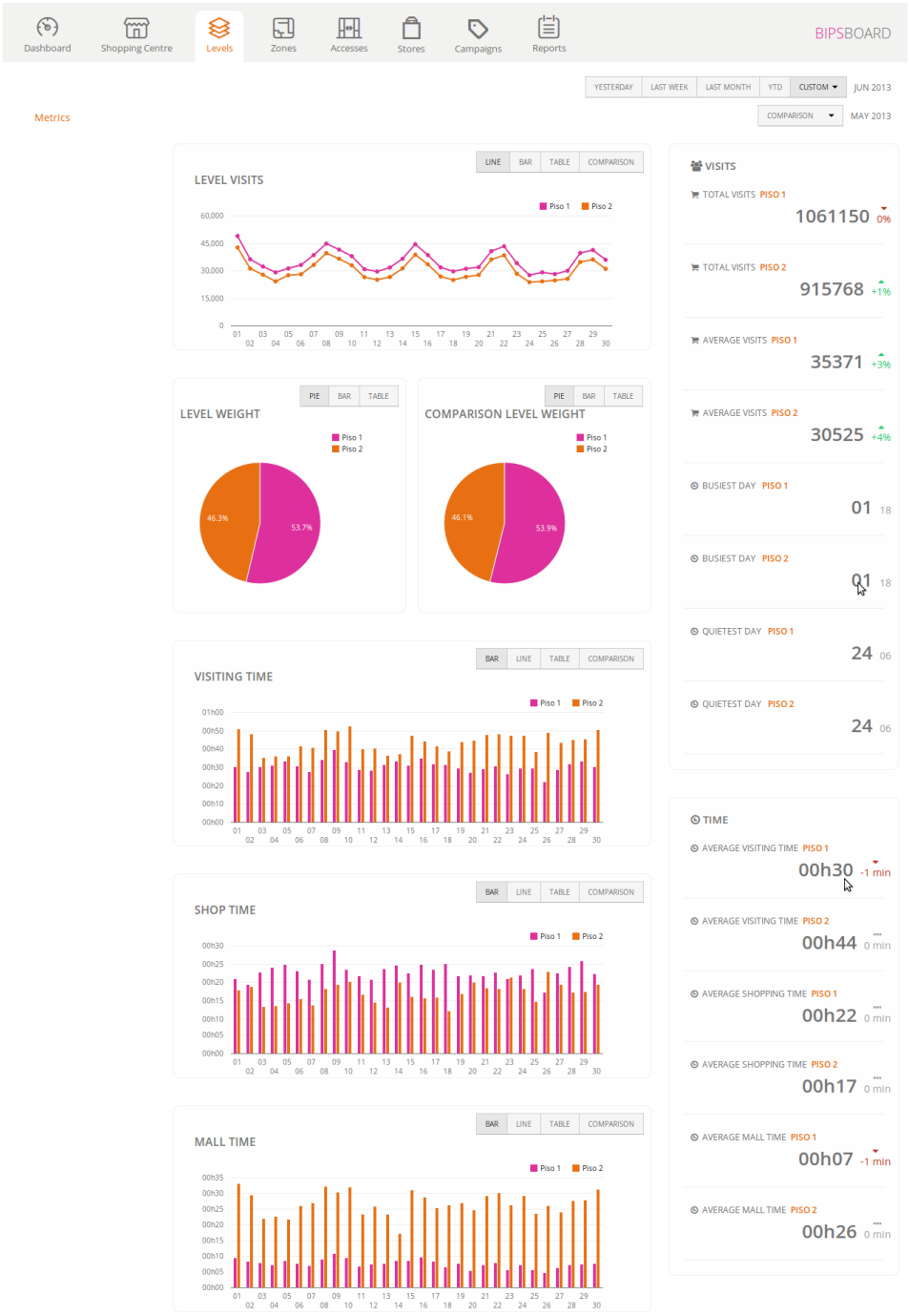

Figura 6.5: Métricas das visitas por piso no mês de Junho

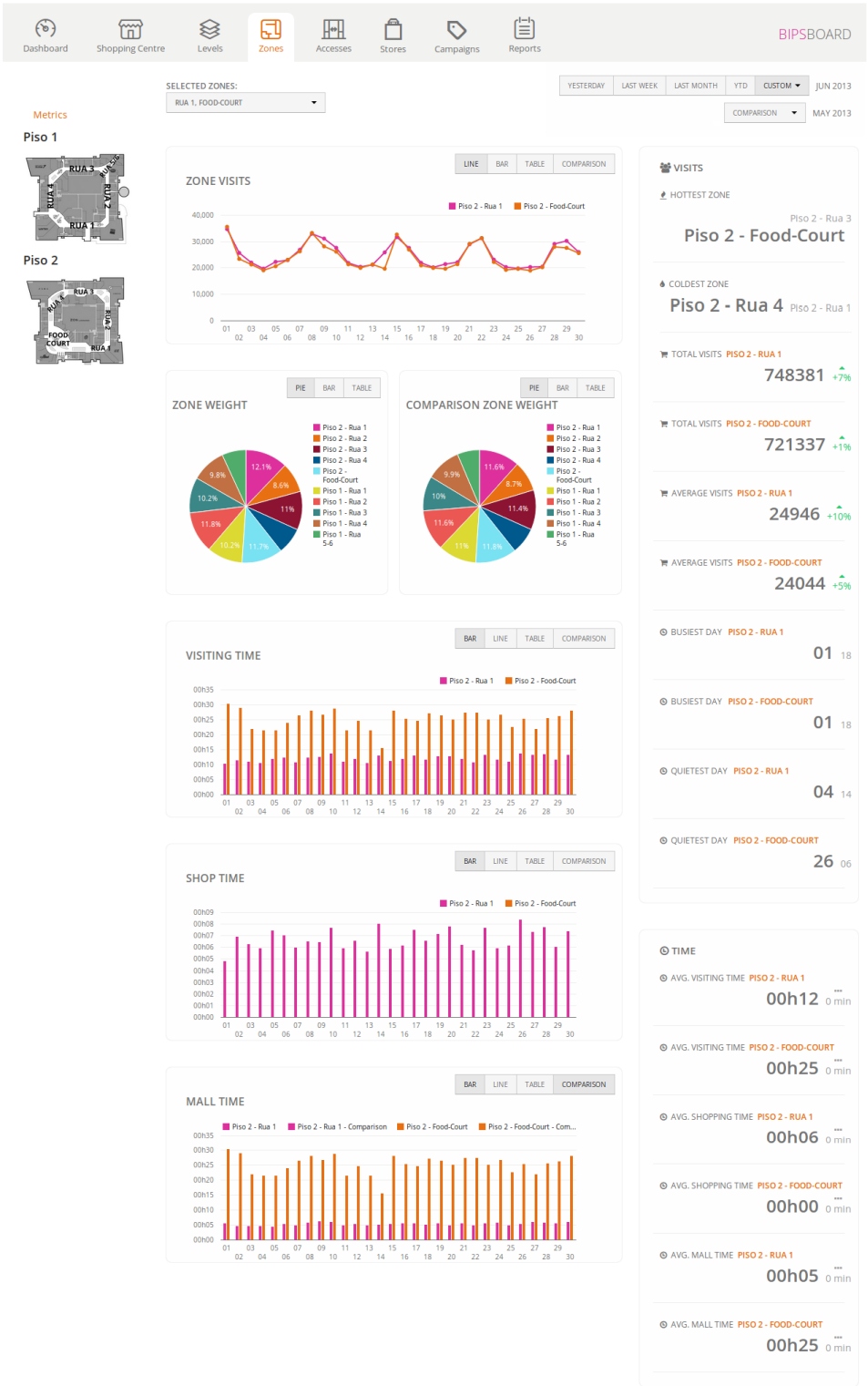

Figura 6.6: Métricas das visitas por Zona no mês de Junho

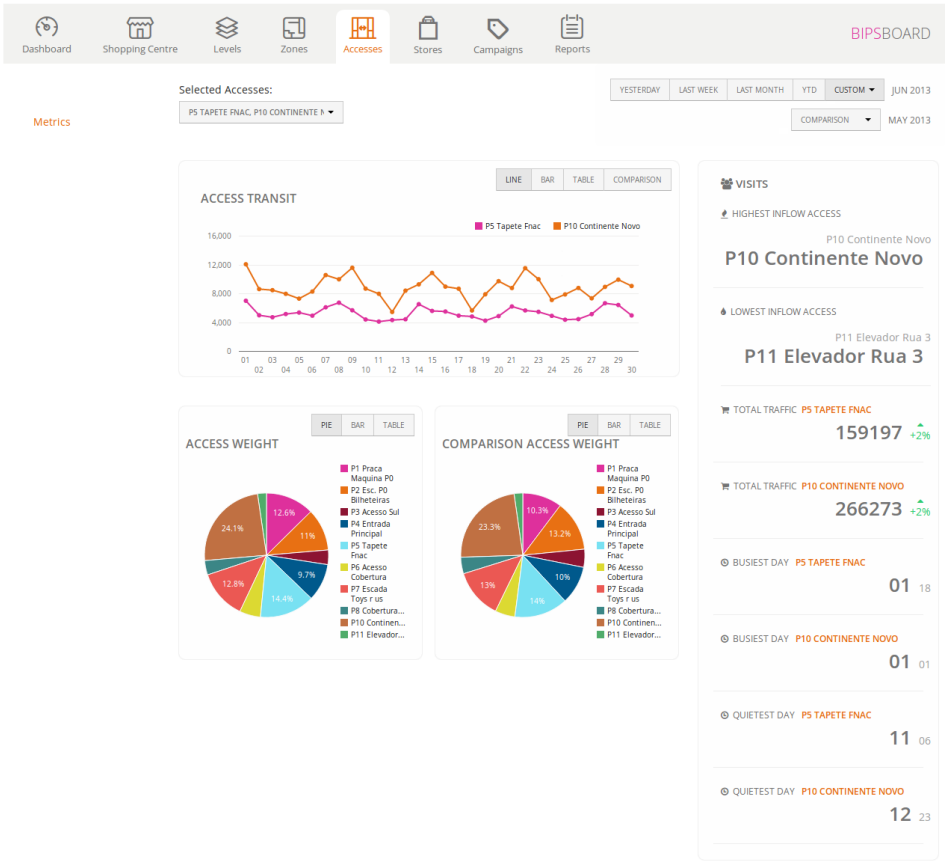

Figura 6.7: Métricas das visitas por Acesso no mês de Junho

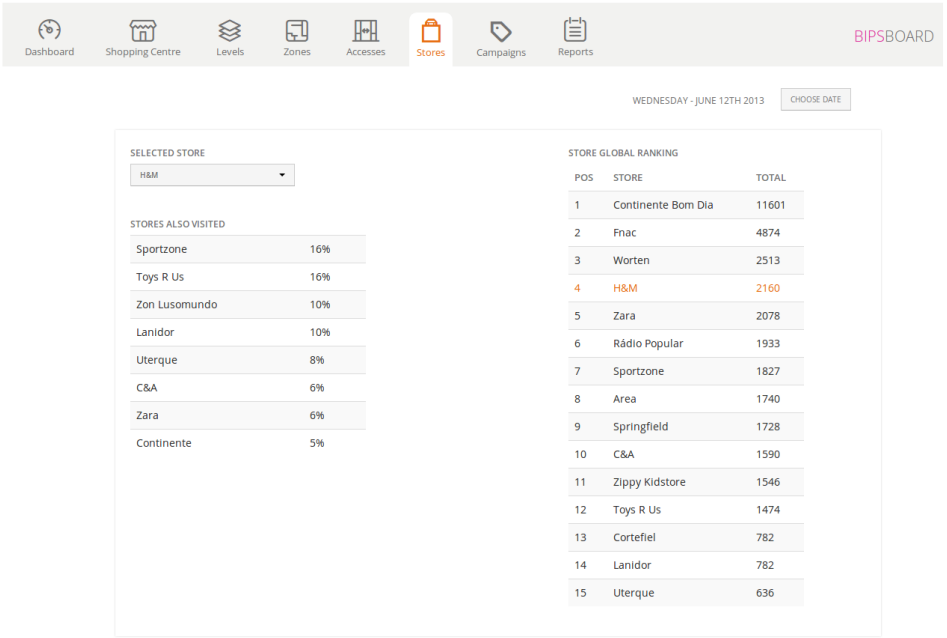

Figura 6.8: Posição das Lojas segundo o número de visitantes no mês de Junho

Capítulo 7

Conclusão

Os objetivos propostos no início do estágio foram atingidos tendo o conjunto de ferramentas de análise de posicionamento sido implementado. Estas ferramentas podem ser usadas para responder às questões colocadas pelo cliente, permitindo aos gestores obterem conhecimento mais preciso sobre o funcionamento do espaço que gerem.

Para conseguir atingir estes objectivos foram realizadas as seguintes tarefas:

- Recolha e compreensão das perguntas de negócio do cliente;
- Conecção à Base de Dados;
- Transformação e pré-processamento dos dados;
- Estudo e aplicação de algoritmos de *Data Mining*;
- Análise dos dados utilizando estatística descritiva para a obtenção de tabelas, grafos e gráficos;
- Seleção com o cliente das ferramentas relevantes para o negócio;
- Integração das ferramentas selecionadas na plataforma BIPS.

Recolha e compreensão das perguntas de negócio do cliente: Foi necessário perceber o negócio do cliente e o modelo de negócio do BIPS. Tornou-se imprescíndivel perceber correctamente cada uma das questões colocadas pelo cliente por forma a transpôlas numa pergunta/tarefa de análise de dados.

Conecção à Base de Dados: Foi necessário estudar a estrutura da BD, que sofreu algumas alterações durante o período inicial de estágio, estudar e escolher a biblioteca de ligação do python à BD e criar a ligação ao cassandra.

Transformação e pré-processamento dos dados: Para se poder trabalhar com os dados foi necessário definir estruturas locais para o seu armazenamento para que o tratamento fosse mais célere e direto. Esta transformação distribui os dados por viagens. Estes dados passaram então por um pré-processamento de modo a se conseguir encontrar o caminho mais frequentado.

Estudo e aplicação de algoritmos de *Data Mining*: O estudo de algoritmos de *Data Mining* foi efectuado com o intuito destes serem aplicados na procura do caminho mais frequente. A possibilidade de os aplicar aos dados obtidos existiu mas, devido a restrições temporais não houve a possibilidade de os incorporar nas ferramentas. Assim foi criado o algoritmo *Paths* [ver algoritmo 5.6.1], que embora mais específico, permite atingir os resultados pretendidos.

Análise dos dados utilizando estatística descritiva para a obtenção de tabelas, grafos e gráficos: Foram aplicadas técnicas de estatística descritiva com a intenção de ver respondidas algumas das questões colocadas pelo cliente.

Seleção com o cliente das ferramentas relevantes para o negócio: Após a aplicação do conjunto de ferramentas a uma amostra de dados, a AK e o cliente procederam a uma análise da informação obtida. Esta análise permitiu saber se existiam questões cujas respostas não proporcionavam uma vantagem para o cliente. Três das ferramentas não foram levadas em conta (mencionadas no capítulo 5). Estas ferramentas já estão prontas e no caso em que um outro cliente da AK esteja interessado, apenas existe a necessidade de as activar.

Integração das ferramentas selecionadas na plataforma BIPS: Chegados ao momento em que sabemos qual o conjunto de ferramentas a utilizar, bastou activá-las e permitir a sua execução uma vez ao dia. Desta forma os gestores do CC obtêm toda a informação relevante ao dia anterior, à semana anterior ou mesmo ao ano anterior.

7.1 Trabalho Futuro

A AK desenvolveu a plataforma BIPS com o objectivo de proporcionar uma solução analítica para auxiliar, não apenas os CC mas também as cadeias de lojas e os

gestores de espaços ao ar livre como Jardins Zoológicos, Parques Temáticos, Feiras etc. Futuramente, o BIPS será desenvolvido para alcançar uma maior adaptabilidade às necessidades dos clientes, melhorando a sua implementação e performance dos algoritmos. Com estes desenvolvimentos procederemos ao estudo de algoritmos e sua implementação para alcançarmos previsões de futuras visitas.

Apêndice A

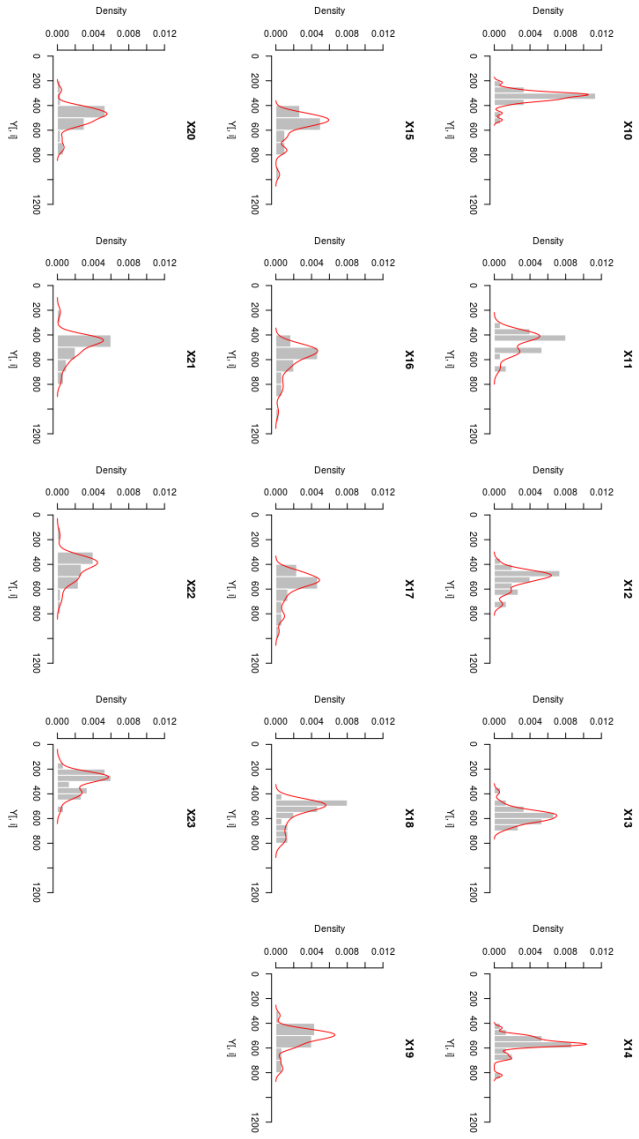

Figura A.1: Densidade das visitas por hora no mês de Junho

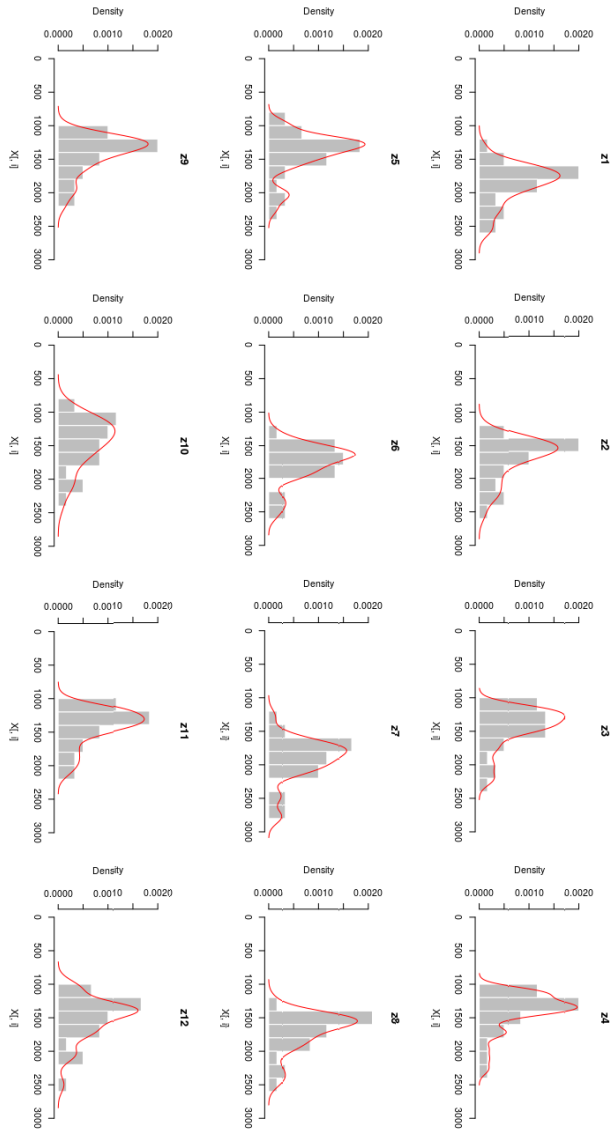

Figura A.2: Densidade das visitas por zona no mês de Junho

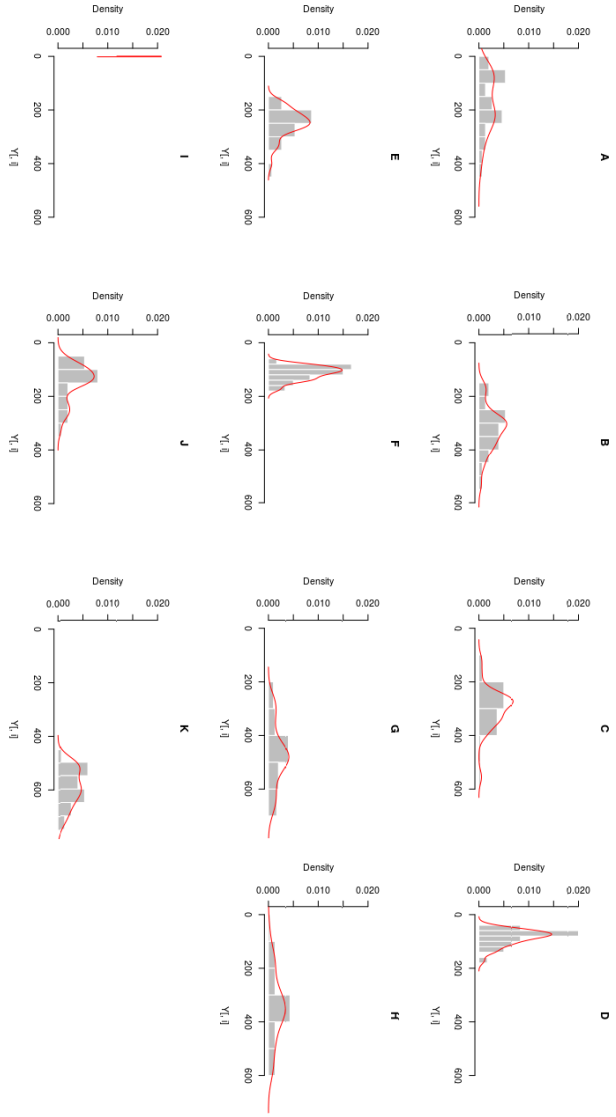

Figura A.3: Densidade das visitas por Acesso no mês de Junho

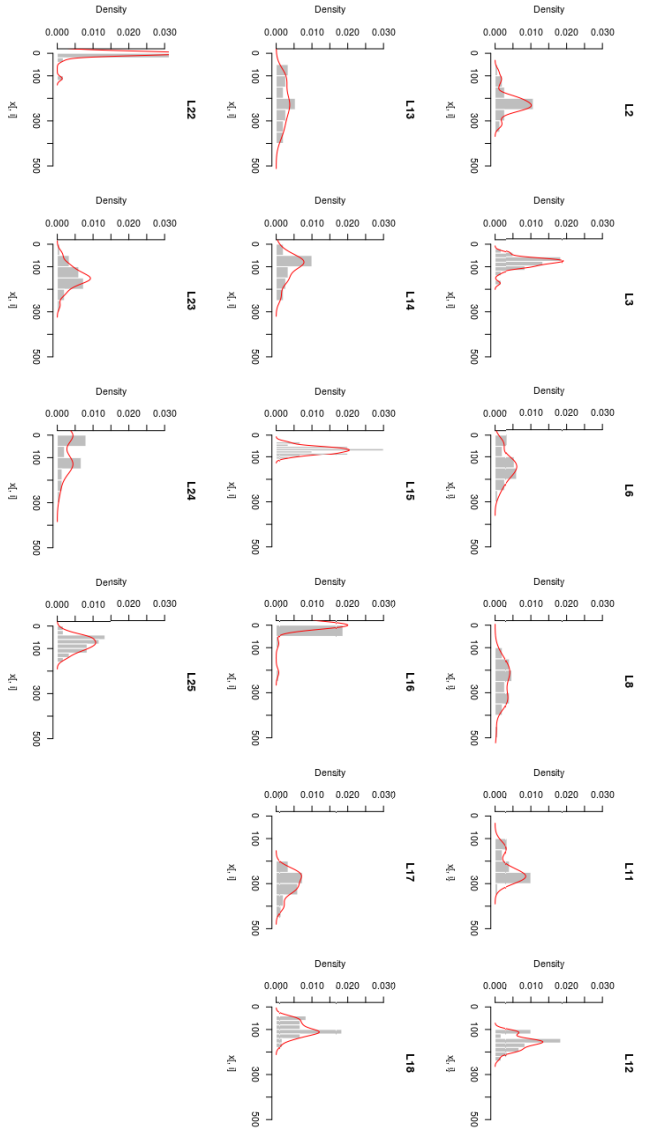

Figura A.4: Densidade das visitas por primeira paragem no mês de Junho

Bibliografia

[3VR] 3vr - video intelligence platform. http://www.3vr.com/. acedido a 12 de Fevereiro de 2013.

[AFGY02] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 429–435, New York, NY, USA, 2002. ACM.

[AS95] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.

[AS08] Ana Azevedo and Manuel Filipe Santos. Kdd, semma and crisp-dm: a parallel overview. In Ajith Abraham, editor, *IADIS European Conf. Data Mining*, pages 182–185. IADIS, 2008.

[Cas] Cassandra. http://cassandra.apache.org/. acedido a 3 de Dezembro de 2012.

[Cor11] Copyright IBM Corporations. Ibm spss modeler crisp-dm guide. Technical report, IBM, 2011.

[Foo] Experian footfall. http://www.footfall.com/. acedido a 12 de Fevereiro de 2013.

[FV13a] Philippe Fournier-Viger. Spmf - sequential pattern mining framework. http://www.philippe-fournier-viger.com/spmf/, 2008 a 2013.

[FV13b] Philippe Fournier-Viger. Spmf - sequential pattern mining framework. http://www.philippe-fournier-

viger.com/spmf/index.php?link=performance.php, 2008 a 2013. 2012-11-28 Which sequential pattern mining algorithm is the most efficient (PrefixSpan vs SPAM)?

[GIea97] Robert Gentleman, Ross Ihaka, and et al. The r project for statistical computing. http://www.r-project.org/, desde 1997. começou no Departamento de Estatística da Universidade de Auckland e foi acedido a 3 de Dezembro de 2012.

[Git12a] Inc. GitHub. github. https://github.com/digg/lazyboy/branches, 2012. acedido a 20 de Dezembro de 2012.

[Git12b] Inc. GitHub. github. https://github.com/pycassa/pycassa/branches, 2012. acedido a 20 de Dezembro de 2012.

[Kno13] Around Knowledge. Bips. http://www.linkedin.com/company/around-knowledge/bips-tagless-real-time-location-intelligence-470175/product?trk=biz$_p$*roduct*, 2013. *acedidoa*11*deMaiode*2013.

[MHW09] Peter Reutemann Mark Hall, Eibe Frank and Ian H. Witten. The weka data mining software. http://www.cs.waikato.ac.nz/ml/weka/, desde 2009. SIGKDD Explorations.

[MZC01] C.L. Yip M. Zhang, B. Kao and D. W. Cheung. A gsp-based efficient algorithm for mining frequent sequences. In *Proceedings of the International Conference on Artificial Intelligence*, IC-AI'2001, Las Vegas, Nevada, 2001.

[PP01] Q. Chen Pei, J. Han and H. Pinto. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering*, ICDE'2001, Washington, DC, USA, 2001.

[RI] Rapid-I. Rapid-i - report the future. http://rapid-i.com/.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, London, UK, UK, 1996. Springer-Verlag.

[Urb12a] Simon Urbanek. *Package 'RCassandra' - Provides access to databases through the JDBC interface*, Dezembro 2012. acedido a 14 de Dezembro de 2012.

[Urb12b] Simon Urbanek. *Package 'RJDBC' - R/Cassandra interface*, Novembro 2012. acedido a 14 de Dezembro de 2012.